## Human-Computer Interaction

<table>
<thead>
<tr>
<th>Exploring the terrain and seeking quick inspiration</th>
<th>Recruiting research participants</th>
<th>Obtaining deep insights into people’s lives and everyday contexts</th>
<th>Eliciting feedback on concepts, insights, or value propositions</th>
<th>Analyzing and synthesizing research findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typeform</td>
<td>Clarity</td>
<td>24tru*</td>
<td>Usabilla</td>
<td>Murally</td>
</tr>
<tr>
<td>Visually engaging, easily shareable surveys</td>
<td>Find practitioner-experts in a given business area</td>
<td>Remote, video-based diary studies and web-based discussions</td>
<td>Visual feedback on web prototypes or live websites</td>
<td>Visual, collaborative organization of material</td>
</tr>
<tr>
<td>Betterific</td>
<td>PivotPlanet</td>
<td>Revelation*</td>
<td>Optimizely</td>
<td></td>
</tr>
<tr>
<td>Crowd sourced inspiration</td>
<td>Find practitioner-experts in a given business area</td>
<td>Remote diary studies, live video and web-based discussions</td>
<td>A/B and multivariate testing platform</td>
<td></td>
</tr>
<tr>
<td>Voice</td>
<td>Ethnio</td>
<td>Crowdtap*</td>
<td>Unbounce</td>
<td></td>
</tr>
<tr>
<td>Quick input and inspiration from voice polls</td>
<td>Intercept website or app users in the moment</td>
<td>Crowd sourced inspiration</td>
<td>Concepts or value prop. testing on landing pages</td>
<td></td>
</tr>
</tbody>
</table>

**model-based user interaction in the context of software engineering**

*We found these platforms to be useful for more than one research activity. For example, Crowdtap is useful for all five research phases.*

[https://labs.ideo.com/2014/09/19/digital-tools-for-design-research/](https://labs.ideo.com/2014/09/19/digital-tools-for-design-research/)
“If you are not embarrassed by the first version of your product, you’ve launched too late.”

Reid Hoffman
How we can evaluate the usability (quality) of interactive systems?
UI quality

Presentation of information

multiplicity of devices & representations
UI quality

Presentation of information

input/output reusability

use output produced by one action as input for another
UI quality

Ordering of task planning

Multiplicity of user roles

Regular user versus administrator
UI quality

Ordering of task planning

multiplicity of execution paths

menu versus toolbox versus shortcuts
UI quality

Ordering of task planning

non-preemptiveness

degree of freedom for users to decide what’s next
UI quality

Ordering of task planning

reachability

possibility to navigate in the system (e.g., undo, redo, breadcrumb pattern)
UI quality

Ordering of task planning

observability *versus* browsability
UI quality

Adaption of interaction

reconfigurability

system ability to support user personalization
UI quality

Adaption of interaction

reconfigurability

system ability to support user personalization

configuration versus personalization

affects system function & performance

relevant to a certain user
UI quality

Adaption of interaction

adaptivity

system ability to support automated adaptation
UI quality

Adaption of interaction

migrability

system ability to transfer responsibility from one user to another, among users, among users and systems/platforms
UI quality

Adaption of interaction

plasticity

system ability to adapt to the context of use while preserving predefined usability properties
UI quality

Standards

ISO/IEC 9126-3 (2003) – internal measures
UI quality

Quality in use is the user’s view of the quality of the software product when it is used in a specific environment and a specific context of use.

UI quality

External quality is the totality of characteristics of the software product from an external view

UI quality

External quality is the totality of characteristics of the software product from an external view measured and evaluated in the testing phase within a simulated environment, by using external metrics
UI quality

**Internal quality** is the totality of attributes of a product that determine its ability to satisfy stated and implied needs when used under specified conditions.

UI quality

**Internal quality** is the totality of attributes of a product that determine its ability to satisfy stated and implied needs when used under specified conditions

remains unchanged until the system redesign
UI quality

For details, consult:

Usability evaluation

Criteria
Scapin & Bastien, 1997; Vanderdonckt, 1995

compatibility
consistency
work load
adaptation
dialog control
guidance
error management
Usability evaluation

Important aspects

*a priori* and/or *a posteriori*

design *versus* testing (evaluation)
Usability evaluation

Example:

the lack of explanatory messages in the case of Web links

Click [here](#) to go to the UAIC main page.
Click [here](#) to visit our HCI Website.
Click [here](#) for details about this event.
Usability evaluation

Example:

the lack of explanatory messages in the case of Web links

Click here to go to the UAIC main page.
Click here to visit our HCI Website.
Click here for details about this event.
Usability evaluation

Example:

adoption of different placement strategies for content and navigational methods
“Optimal” placement of links for the main page of a Website (Shaikh & Lenz, 2006)
“Optimal” placement of links for the subsidiary pages (Shaikh & Lenz, 2006)
“Optimal” placement of the internal search tools (Shaikh & Lenz, 2006)
“Optimal” placement of the advertisements (Shaikh & Lenz, 2006)
Usability evaluation

One of the most used general technique: eye tracking

30—60 minute interviews, in which users are asked to complete real-life tasks online, while the eye tracking monitor captures their eye movements


www.nngroup.com/reports/how-to-conduct-eyetracking-studies/
What areas of the page draw users’ attention?
Do users notice and use key navigation elements?
Do users notice key marketing elements and do they recall them?
Are users successful in completing a particular task?

www.evocinsights.com
Which pages, ads, videos, or images do users prefer?
What do users expect to find but overlook on the site?
Usability evaluation

One of the most used general technique: eye tracking

several open-source software solutions:
EyeWriter – http://eyewriter.org/developer/
Gaze Tracking – sourceforge.net/projects/gazetrackinglib/
OGAMA – http://www.oggama.net/
PyGaze – http://www.pygaze.org/
Usability evaluation

PUI – Plastic User Interfaces

adaptation to the context of use while satisfying predefined usability properties of interest
D. Thevenin, J. Coutaz & G. Calvary, 2004
Usability evaluation

PUI – Plastic User Interfaces

levels of adaptation:
  lexical
  sintactic
  semantic
performed tasks concerning the user goals
Accessibility

successful access to information and IT applications by people having special needs

visual, hearing, motor, cognitive, seizure disorders
Accessibility

realities:
20% of US population have a certain type of disability
10% of persons are having severe problems

4% of world-wide humans have major sight problems
general usability affects all users

physical barriers affects only disabled people

variables inconveniences for some barriers for others
**discussion**

**Touch**
- one arm
- arm injury
- new parent

**See**
- blind
- cataracts
- distracted driver
discussion

Hear
- deaf
- ear infection
- bartender

Speak
- non-verbal
- laryngitis
- heavy accent
cognitive issues
Accessibility

offering alternative means in order to facilitate the completion of users’ tasks for people having temporary (on short/long term) or permanent problems
Accessibility

general strategies:

textual descriptions of multimedia content
(images, audio, animations, video, 3D)
Accessibility

general strategies:

a proper (logical) organization of the content and navigational paths
Accessibility

general strategies:

support for keyboard-only interaction
Accessibility

general strategies:

using standardized formats
Accessibility

guidelines and tools – concerning Web applications:

www.w3.org/WAI/

www.webaim.org
WAVE (Web Accessibility Evaluation Tool)
Accessibility

see also Ire Aderinokun, *The Accessibility Cheatsheet* (2015)

https://bitsofco.de/the-accessibility-cheatsheet/
Internationalization (I18N) and localization (L10N)

*language*
linguistic preferences of the users

*locale*
cultural preferences concerning number and date formatting, currency, etc.
Internationalization (I18N) and localization (L10N)

“If the user can’t read the description of the preference, he/she doesn’t even have a chance to make a choice.”

Achim Ruopp, 2007
Internationalization (I_{18N}) and localization (L_{10N})

it is desirable to use complete phrases in order to give a better translation
Internationalization (I18N) and localization (L10N)

Web guidelines & resources:

www.w3.org/International/

www.globalbydesign.com/
avoid absolute width (texts in other languages could have different lengths)

avoid pictures (use translatable text)
Usability evaluation (Rogers, 2007)

analytical evaluation
controlled experiment
field study
formative evaluation
heuristic evaluation
predictive evaluation
summative evaluation
usability laboratory
usability studies
user testing
Multiple existing heuristics introduced by Jakob Nielsen (1994)

www.nngroup.com/articles/ten-usability-heuristics/
Multiple existing heuristics proposed by Bruce Tognazzini (revised in 2014)

<table>
<thead>
<tr>
<th>Aesthetics</th>
<th>Discoverability</th>
<th>Metaphors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anticipation</td>
<td>Efficiency of the User</td>
<td>Protect Users’ Work</td>
</tr>
<tr>
<td>Autonomy</td>
<td>Explorable Interfaces</td>
<td>Readability</td>
</tr>
<tr>
<td>Color</td>
<td>Fitts’ Law</td>
<td>Simplicity</td>
</tr>
<tr>
<td>Consistency</td>
<td>Human-Interface Objects</td>
<td>State: Track it</td>
</tr>
<tr>
<td>Defaults</td>
<td>Latency Reduction</td>
<td>Visible Interfaces</td>
</tr>
<tr>
<td></td>
<td>Learnability</td>
<td></td>
</tr>
</tbody>
</table>

http://asktog.com/atc/principles-of-interaction-design/
Multiple existing heuristics
context: mobile computing (E. Bertini et al., 2006)

1. Visibility of system status & device findability
2. Match between system and the real world
3. Consistency and mapping
4. Good ergonomics & minimalist design
5. Ease of input, screen readability and glancability
6. Flexibility, efficiency of use and personalization
7. Aesthetic, privacy and social conventions
8. Realistic error management
Usability evaluation (Rogers, 2007)

The evaluator could not substitute the real users

an UI has usability problems
only if its final common users have these problems

code inspection vs. code testing
User testing

UI quality evaluation could be performed by users
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

the focus is on usability
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

the participants are (potential) end users
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

there is a real product/service to be evaluated
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

the participants perform tasks, usually while thinking aloud
User testing

Aspects of interest (J. Dumas & J. Fox, 2008)

the data are recorded and further analyzed
a usability test plan HTML5 template

http://profs.info.uaic.ro/~stefan.negru/usability/
User testing

Considered metrics

effectiveness, efficiency & satisfaction
User testing

Considered metrics
effectiveness, efficiency & satisfaction

+ 

fun, challenge & stimulation
User testing: planning (Adaptive Path, 2001)

$t - 2$ weeks  Determine test audience, start recruiting immediately

$t - 2$ weeks  Determine feature set to be tested

$t - 1$ week  Write first version of guide, discuss with team, check on recruiting

$t - 3$ days  Write second version of guide, recruiting should be completed

$t - 2$ days  Complete guide, schedule practice test, set up and check equipment

$t - 1$ day  Do practice test in the morning, adjust guide/tasks as appropriate

$t$  Test (usually 1-2 days, depending on scheduling)

$t + 1$ day  Discuss with observers, collect copies of all notes

$t + 3$ days  Watch all video recordings, take notes

$t + 1$ week  Combine notes, write analysis

$t + 1$ week  Present to team, discuss and note directions for further research
User testing

Methods:

- formative evaluation
- field study
- controlled experiment
User testing

Formative evaluation

finding problems for next iteration of the design project
User testing

Formative evaluation

prototype/implementation is evaluated within a controlled environment (lab), with focus on specific tasks
User testing

Formative evaluation

users, facilitators, observers offers qualitative data (usability issues)
User testing

Field study

tries to find problems with respect to a given context
User testing

Field study

evaluates the (preliminary) UI in a concrete context, with focus on real tasks
User testing

Field study

evaluates the (preliminary) UI in a concrete context, with focus on real tasks

offers qualitative annotations
User testing

Controlled experiment
to test a hypothesis

e.g., interface $X$ is easier to be used than interface $Y$
User testing

Controlled experiment

could be used to evaluate a preliminary UI, in lab rigorous conditions, with focus on specific tasks

has one or more **conditions** (independent variables) and **measures** (dependent variables)
User testing

Controlled experiment gives quantitative information:
  time of reaction, error rate, user satisfaction,...

exp-platform.com/Documents/GuideControlledExperiments.pdf
User testing

Advices + aspects of interest (Lukas Mathis, 2011)

- do not influence the tester
- avoid stressful situations
- consider the ethics of conducting tests
User testing

A/B testing

comparing 2 versions of an UI element or an entire Web page in a time interval to see which performs better
User testing

A/B testing

comparing 2 versions of an UI element or an entire Web page in a time interval to see which performs better

performance criterion = conversion rate (from visitors to goal achievers)
User testing

www.alistapart.com/articles/a-primer-on-a-b-testing/

see also http://elem.com/~btilly/effective-ab-testing/
User testing

A/B testing

multivariate testing – different versions of individual parts of the design are tested at the same time
a multivariate test (Julian Gaviria, 2017)

**Discussion**

**Variant A**
- Single Offer with Intro Message

**Variant B**
- Top 3 Converting eBooks

**Variant C**
- Single Offer w/ Intro Msg + Top 2 eBooks

Website Converted at **8.59%** with 108 leads generated out of 1,258 visitors (**+22.3% Improvement**).

Website Converted at **8.14%** with 109 leads generated out of 1,339 visitors (**+16.0% Improvement**).

Website Converted at **10.12%** with 127 leads generated out of 1,255 visitors (**+44.2% Improvement**).

https://julian.is/article/exit-intent-popups/
User testing

A/B testing

several tools:
- Convert
- Optimizely
- Unbounce
- Vanity – http://vanity.labnotes.org/
- Visual Website Optimizer
<table>
<thead>
<tr>
<th><strong>Majesticwine</strong></th>
<th><strong>201%</strong> increase</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Redesigning category webpage increases leads generated</strong></td>
<td></td>
</tr>
<tr>
<td>Majestic Wines revamped their category page design to increase online enquiries for their Wedding services by 201%.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>server density</strong></th>
<th><strong>144%</strong> increase</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A/B testing between different pricing structures increases revenue by 114%</strong></td>
<td></td>
</tr>
<tr>
<td>Server Density A/B tested between per unit and packaged pricing plans. The winning plan reduced free signups but increased the Average Order Value (AOV), and consequently revenue by 114%.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Taylor Gifts</strong></th>
<th><strong>111%</strong> increase</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Redesign of ecommerce product page increases conversions</strong></td>
<td></td>
</tr>
<tr>
<td>Conversion Optimization Agency Trinity Insight used Visual Website Optimizer to test a better version of the ecommerce product page. This led to a 111% increase in conversions.</td>
<td></td>
</tr>
</tbody>
</table>
User testing

Pilot study

a trial run of an experimental procedure, not expected to produce valid research data
User testing

Remote testing

by using a screen-sharing software

User testing

Remote testing

the tester’s environment cannot be typically controlled

also, the tester might get distracted during the test
Predictive evaluation

Having a correct model of the interaction between users and computers, we can predict the usability of a system, without the need of designing and/or testing a concrete UI.
Predictive evaluation

User testing could only relieve certain problems, but cannot explain them.
Predictive evaluation

Classical models:

- probabilistic – e.g., Bayes networks
- GOMS (Goals, Operators, Methods, Selection rules)
- CMP-GOMS (Cognitive-Motor-Perceptual)
task graph used by CMP-GOMS model

the critical path is the path of tasks having the longest time of execution
Predictive evaluation

Directions of research

objectives beyond productivity

- **hedonomics** (Hancock *et al.*, 2005)
Predictive evaluation

Directions of research

RITE – Rapid Iterative Test and Evaluation

used by Microsoft (M. Medlock et al., 2002, 2005)

focused on fixing usability problems rather than finding them
Usability evaluation

For further information, study:

[www.interaction-design.org/encyclopedia/usability_evaluation.html](http://www.interaction-design.org/encyclopedia/usability_evaluation.html)

Reality

Major differences between software engineering and the effective UI development
Reality

Target-applications & domains of interest

personal productivity, business, entertainment, etc.
Reality

Notations and engineering tools

(formal) methods for software development
Reality

Multiple interaction paradigms

textual, graphical, multimedia, natural, 3D,...
Evolution of the interactive apps: context of use = (U, P, E)

Jean Vanderdonckt, 2006
Reality

Multiple models to be considered (Abrams et al., 2001)

<table>
<thead>
<tr>
<th>Application 1</th>
<th>Platform #1</th>
<th>Platform #2</th>
<th>Platform #3</th>
<th>Platform #4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UI #1</td>
<td>UI #2</td>
<td>UI #3</td>
<td>UI #4</td>
</tr>
<tr>
<td>Application 2</td>
<td>UI #5</td>
<td>UI #6</td>
<td>UI #7</td>
<td>UI #8</td>
</tr>
<tr>
<td>Application 3</td>
<td>UI #9</td>
<td>UI #10</td>
<td>UI #11</td>
<td>UI #12</td>
</tr>
</tbody>
</table>
Reality

Multiple models to be considered (Abrams et al., 2001)

<table>
<thead>
<tr>
<th>Application 1</th>
<th>Platform #1</th>
<th>Platform #2</th>
<th>Platform #3</th>
<th>Platform #4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application 2</td>
<td>UI #1</td>
<td>UI #2</td>
<td>UI #3</td>
<td>UI #4</td>
</tr>
<tr>
<td>Application 3</td>
<td>UI #5</td>
<td>UI #6</td>
<td>UI #7</td>
<td>UI #8</td>
</tr>
<tr>
<td>Application 3</td>
<td>UI #9</td>
<td>UI #10</td>
<td>UI #11</td>
<td>UI #12</td>
</tr>
</tbody>
</table>

Application 1 → UI model #1 → Platform model #1 → Platform #1
Application 2 → UI model #2 → Platform model #2 → Platform #2
Application 3 → UI model #3 → Platform model #3 → Platform #3

Dr. Sabin-Corneliu Buraga – profs.info.uaic.ro/~busaco/
Model-based UI design

MDA (Model-Driven Architecture)

www.omg.org/mda/
Model-based UI design

Model(s) facilitate(s) an abstract view of the interaction

separation of concerns, ability of correlation

parsability, editability

if possible, human readability
Model-based UI design

Models

explicitly capture knowledge about UI and interactive applications with appropriate abstractions
Model-based UI design

Method

structures the definition and use of underlying models in a stage-wise approach

example: agile usability (Scott Ambler, 2008)

www.agilemodeling.com/essays/agileUsability.htm
Model-based UI design

Supporting tools

support the use of the method by providing tools for models and their related operations

ideally, one model should be supported by at least one tool
Models used in the processes of UI design
(P. Forbrig et al., 2004)
Using notations to model tasks to be performed by users (L. Marucci et al., 2004)
Process Reference Framework for the development of plastic user interfaces
S = Source context of use

T = Target context of use

User S, Platform S, Environment S

Task and Domain S

Abstract user Interface S

Concrete user Interface S

Final user Interface S

User T, Platform T, Environment T

Task and Domain T

Abstract user Interface T

Concrete user Interface T

Final user Interface T

Jean Vanderdonckt, 2006
Bouillon (2006)
Model-based UI design

CADUI – Computer-Aided Design of User Interface

(formal) descriptions of interactive systems, in terms of existing meta-models, used to design and deploy multiple user interfaces
Model-based UI design

Jean Vanderdonckt
Model-based UI design

At the abstract level, the specification of the interactive system is given by CIM (Computation-Independent Model)
Model-based UI design

At the abstract level, the specification of the interactive system is given by CIM (Computation-Independent Model).

- Mission statement
- Function reference tree
- Use cases
- Interaction requirements
- Concurrent tasks trees
Model-based UI design

At the abstract level, the specification of the interactive system is given by CIM (Computation-Independent Model)

see also “Requirements Engineering from an HCI Perspective” (A. Sutcliffe, 2015)
http://www.interaction-design.org/encyclopedia/requirements_engineering.html
Model-based UI design

The platform independent description is based on PIM (Platform-Independent Model)
Model-based UI design

The platform independent description is based on PIM (Platform-Independent Model)

- object model
- dynamic model
- functional model
- presentation model
Model-based UI design

The development focused on a specific platform is based on PSM (Platform-Specific Model)
Model-based UI design

The development focused on a specific platform is based on PSM (Platform-Specific Model)

uses software tools of transformation (compilation) based on an application model
Model-based UI design

The concrete implementation is using CM (Code Model)
Model-based UI design

The concrete implementation is using CM (Code Model)

concerns creation/generation of the source-code:
  interface tier
  application tier
  persistence tier
  integration
Model-based UI design

Task and Domain → Abstract User Interface → Concrete User Interface → Final User Interface

T1 Rendering

T3

Abstract User Interface
Model-based UI design

Task and Domain

Abstract User Interface

Concrete User Interface 1
(2-D Desktop)

Rendering

Final User Interface

Concrete User Interface 2
(2-D small display)

Rendering

Final User Interface

Concrete User Interface 3
(auditory)

Rendering

Final User Interface

Task and Domain

Abstract User Interface

Concrete User Interface

Rendering

Final User Interface

Task and Domain

Abstract User Interface

Concrete User Interface

Rendering

Final User Interface
Model-based UI design

(Meta)languages for describing UI

UIML – User-Interface Modeling Language
UsiXML – USer Interface eXtensible Markup Language
XAML – eXtensible Application Markup Language
XUL – eXtensible User-interface Language
Model-based UI design

MDA (Model-Driven Architecture)

- Computing Independent Model (CIM)
- Model to Model
- Platform Independent Model (PIM)
- Model to Model
- Platform Specific Model (PSM)
- Model to Code
- Source code

UsiXML

- UsiXML models: task, domain
- Graph transformations
- UsiXML model: Abstract user interface
- Graph transformations
- UsiXML model: Concrete user interface
- Rendering
- Final user interface

Jean Vanderdonckt, 2006
designing the abstract UI of a computer game
user task modelling via a specific tool
(S. Tofan, A. Pradais, S. Buraga, 2009)
Support for multiple displays
(Grolaux & Vanderdonckt, 2005)

detach ▶ migrate ▶ plastify
case study
Support for multiple displays
(Grolaux & Vanderdonckt, 2005)

detach ▶ migrate ▶ plastify

for migration, rules and/or design patterns could be used
Support for multiple displays
(Grolaux & Vanderdonckt, 2005)

the use of rules – example:

\[ \forall x \in T_s : x = \text{input and } (x\.\text{type} = \text{"text" or } x\.\text{type} = \text{"password" or } x\.\text{type} = \text{NULL}) \rightarrow \text{AddNode ("textComponent", idText) where idText = NodeValue (T_t)} \]
case study

substitution  removal  moving

examples of actions that could be performed for migration and/or creating PUI – revisit responsive Web design
Model-based UI design

For other examples, read G. Meixner, G. Calvary, J. Coutaz (Eds.), *Introduction to Model-Based User Interfaces*, W3C Working Group Note (2014)

www.w3.org/TR/mbui-intro/
No more programming: only models

Target Applications, Domains

Notations & Engineering Tools

User Interface Interaction Techniques

All Applications

Invisible UI

No more programming: only models

Embedded systems
Command & control systems, games
3D Applications

Physical models
World models
Scene model
Task model, context model, UML,…

Embodied UIs
Tangible UIs
Mixed Reality User Interfaces
Virtual Reality User Interfaces
Multi-platform User Interfaces

Graphical User Interfaces
Character UIs
No Interaction Technique

Web, desktop, mobile apps

Information systems

Business applications
Automated, batch systems

Entity-relationship model
Attribute model
State-transition diagrams
Screen definitions
Nothing

Palanque, 2002 & Vanderdonckt, 2006

2020
“Conclusion”

interactivity in the context of software engineering
UI quality, testing, models

Basecamp — When there is a form field error, the character on the left makes a surprising facial expression.
Pink Floyd
1965 - 1995
Roger Waters, David Gilmour, Rick Wrights, Nick Mason

Song Stats
Us and Them (1973)
Shine on You Crazy Diamond Parts I-V (1975)

next episode: data visualization