All of the other processes are closely linked with evaluation. Systems must be **verified** and **validated**:
- **Verification** is correctness within the terms of product specification
  - Designing the product right
- **Validation** is the correctness of the product within the terms of its intended use
  - Designing the right product

---

### Why Do We Evaluate?

**A. Pre-design**
- what do people do?
- how can we understand what we need in system functionality?

► Evaluation produces:
- key tasks, functional and non-functional requirements
- work practices
- organizational practices
- user types

---

### Why Do We Evaluate?

**B. During initial stage, developing design ideas and representations**
- evaluate choices of initial design ideas and representations
  - is the representation appropriate?
  - does it reflect how people think of their task?

► Evaluation produces:
- user reaction to high-level design
- validation and list of problem areas at conceptual level
- new design ideas

---

### Why Do We Evaluate?

**C. During iterative development, refining a design / representation**
- fine tune the interface, looking for usability bugs
  - can people use this system?

► Evaluation produces:
- user reaction to low-level design
- validation and list of problem areas (bugs)
- variations in design ideas

---

### Why Do We Evaluate?

**D. Post-design**
- **acceptance test**: did we deliver what we said we would?
- **revisions**: what do we need to change?
- **effects**: What did we change in the way people do their tasks?

► Evaluation produces:
- testable usability metrics
- actual reactions
- validation and list of problem areas (bugs)
- changes in original work practices/requirements
Evaluation Process

1. Developing a test plan (“blueprint” for the test)
   - General intention and individual checks
     - MUST BE PRECISE
     - MUST BE CLEAR
     - MUST BE MEASURABLE/OBSERVABLE
   - Specifies the resources needed to carry out test
   - Governs conclusions that can be made
2. Developing test materials
3. Recruiting participants
4. Carrying out the test
5. Debriefing participants
6. Findings and recommendations

Test plan

1. Purpose of test
2. Problem statements/test objectives
3. Participant profile (inclusion/exclusion criteria)
4. Method/technique to be used
5. List of tasks to be used
6. Test environment (field vs. lab) and material (HW/SW, resources, recorder and batteries, report forms, questionnaires)
7. Experimenter’s role (monitor, coach etc.)
8. Evaluation measures to be taken (qualitative vs. quantitative, subjective vs. objective)
9. Contents of report to be produced and how the report is going to be presented → focus group, informal meeting, big bosses are going to be there

Quantitative vs. Qualitative

<table>
<thead>
<tr>
<th>Quantitative</th>
<th>Qualitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers</td>
<td>Words</td>
</tr>
<tr>
<td>Point of view of researcher</td>
<td>Points of view of participants</td>
</tr>
<tr>
<td>Researcher distant</td>
<td>Researcher close</td>
</tr>
<tr>
<td>Theory testing</td>
<td>Theory emergent</td>
</tr>
<tr>
<td>Static</td>
<td>Process</td>
</tr>
<tr>
<td>Structured</td>
<td>Unstructured</td>
</tr>
<tr>
<td>Generalization</td>
<td>Contextual understanding</td>
</tr>
<tr>
<td>Hard, verifiable data</td>
<td>Rich, deep data</td>
</tr>
<tr>
<td>Macro</td>
<td>Micro</td>
</tr>
<tr>
<td>Behavior</td>
<td>Meaning</td>
</tr>
<tr>
<td>Artificial settings</td>
<td>Natural settings</td>
</tr>
</tbody>
</table>

Evaluation measures

<table>
<thead>
<tr>
<th>Subjective</th>
<th>Quantitative</th>
<th>Qualitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many errors do you think you made during the test?</td>
<td>Do you believe you did well or poorly, overall?</td>
<td></td>
</tr>
<tr>
<td>Did you feel stressed during the test?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objective</th>
<th>Quantitative</th>
<th>Qualitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many errors were made during the test?</td>
<td>Was the participant’s performance:</td>
<td></td>
</tr>
<tr>
<td>• Poor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Adequate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Good</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Excellent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Did the participant shake or sweat during the test?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Why Use Different Methods?

- Information requirements differ
  - pre-design, iterative design, post-design, generalizable knowledge...
- Information produced differs
  - outputs should match the particular problem/needs
- Cost/benefit of using a certain method
- One method’s strength can complement another’s weakness
  - no one method can address all situations
- Constraints
  - may force you to chose quick and dirty discount usability methods

How Can We Compare Methods?

- Type of information (qualitative vs. quantitative)
- Relevance
  - does the method provide information to our question / problem?
- Setting
  - is it important that the system be evaluated in-context?
- Generalization
  - how well can I generalize the information produced to other situations?
- Repeatability
  - would the same results be achieved if the test were repeated?
How Can We Compare Methods?
► Quickness
  ▪ can I do a good job with this method within my time constraints?
► Cost
  ▪ is the cost of using this method reasonable for my question?
► Equipment
  ▪ What special equipment / resources required?
► Personnel, training and expertise
  ▪ What people / expertise are required to run this method?
► Validity
  ▪ External validity: can the results be applied to other situations?
  ▪ Internal validity: do we have confidence in our explanation?

How Can We Compare Methods?
► Subject selection
  ▪ how many do I need, who are they, and can I get them?
► Scope of subjects
  ▪ is it good for analyzing individuals? small groups? organizations?
► Control
  ▪ do I need to control for certain factors to see what effects they have?
► Cross-sectional or longitudinal
  ▪ is it important that changes over time are measured?
► Support
  ▪ are there tools for supporting the method and analyzing the data?
► Comparative
  ▪ can I use it to compare different things?

Evaluation approaches
► Two approaches, complementary but different
► Formative → normally qualitative data (strategies, problem descriptions, comments)
  ▪ Carried out early and throughout system development with the goal of guiding design
► Summative → more focused on quantitative data (success rates, times, errors, satisfaction)
  ▪ Carried out at the end or at milestones during system development with the goal of assessing how well the system has met its objectives

Evaluation methods
► Inquiry: Involving real users but not in a laboratory
  ▪ “ecological validity” of evaluation
  ▪ User observation, Focus Groups, Interviews
► Inspection: done by interface professionals, no end users necessary
► Testing: Involving real users in controlled setting

Predictive/Expert Evaluation/Inspection
► Usability experts make judgements
  ▪ Based on experience
  ▪ Guided by ‘rules of thumb’
► Rationale
  ▪ Observing users can be time-consuming and expensive
  ▪ Try to predict usability rather than observing it directly
  ▪ Conserve resources (quick & low cost)
► Expert reviewers often used
  ▪ HCI experts interact with system and try to find potential problems and give prescriptive feedback
  ▪ Best if they
    ▪ Haven’t used earlier prototype
    ▪ Familiar with domain or task
    ▪ Understand user perspectives
Predictive/Expert Evaluation/Inspection

- Feature Inspection
  - System features are inspected in the context of a task (e.g. word processor in the context of writing a letter).
- Perspective-based Inspection
  - Inspecting the system in 3 scenarios: expert users, novice users and error handling.
  - Each inspector works on 1 scenario only.
- Cognitive/Pluralistic Walkthroughs
  - One or more testers "walk" through a task and evaluate system’s usability.
  - Can be either paper mock-up or working prototype.
- User/Predictive Modelling
  - KLM, GOMS, Lee MacGregor model, Fitts’ Law
- Heuristic Evaluation
  - Recognition rather than recall
  - Aesthetic and minimal design
  - Help users recognize, diagnose, and recover from errors
  - Help and documentation
  - Prediction/Expert Evaluation/Inspection

Nielsen’s Heuristics

- Recognition rather than recall
- Flexibility and efficiency of use
  - Shortcut keys
- Aesthetic and minimal design
- Help users recognize, diagnose, and recover from errors
- Help and documentation
- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention

Other Heuristics

- Virtual reality applications heuristics: http://dx.doi.org/10.1016/j.intcom.2004.05.001
- Information visualization heuristics: http://portal.acm.org/citation.cfm?id=1843029
- Web accessibility heuristics: http://www.w3.org/2001/05/hfweb/heuristics.htm

Procedure

1. Gather inputs
   - Familiarize with the domain and task
   - Get the prototype to be studied
2. Evaluate system (individually)
   - Perform two or more passes through system inspecting each screen and flow from screen to screen
   - Evaluate against heuristics
3. Debriefing and collection (collectively)
   - Organize all problems found by different reviewers
   - Decide what are and aren’t problems
4. Report severity rating
   - 0-4 rating scale based on frequency, impact, persistence

Nielsen’s Heuristics

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention

(http://www.useit.com/papers/heuristic/severityrating.html)
Cognitive Walkthrough
► Walkthrough the interface based on a cognitive model
► Evaluation of actions and cues of the interface in comparison to the goals and the background of the typical users
► Process:
  ▪ Construct carefully designed tasks from system spec or screen mock-up
  ▪ Walk through (cognitive & operational) activities required to go from one screen to another
  ▪ Review actions needed for task, attempt to predict how users would behave and what problems they'll encounter

CW: Answering the Questions
1. Will user be trying to produce right effect?
► Typical supporting evidence
  ▪ It is part of their original task
  ▪ They have experience using the system
  ▪ The system tells them to do it
2. Will user notice that correct action is available?
► Typical supporting evidence
  ▪ Experience
  ▪ Visible device, such as a button
  ▪ Perceivable representation of an action such as a menu item

CW: Next Question
3. Will user know it’s the right one for the effect?
► Typical supporting evidence
  ▪ Experience
  ▪ Interface provides a visual item (such as prompt) to connect action to result effect
  ▪ All other actions look wrong
4. Will user understand the feedback?
► Typical supporting evidence
  ▪ Experience
  ▪ Recognize a connection between a system response and what user was trying to do

Usability Evaluation Testing
► Thinking Aloud Protocol
  ▪ Users are asked to verbalize their thoughts, feelings and opinions when using the system.
  ▪ Two variations: critical response & periodic report.
  ▪ Useful to understand user's mental model, interaction with the system and terminology.
► Co-discovery Learning
  ▪ Users work in pair to achieve a common goal with the tested system.
  ▪ Users are encouraged to "think-aloud".
► Coaching Method
  ▪ Participants asked an expert coach.
  ▪ Participant-coach & participant-computer interactions are observed.

Usability Evaluation Testing
► Shadowing method
  ▪ An expert user sits next to a user and explains the user’s behaviour to the usability tester.
  ▪ Usually when it is not appropriate for users to give any response during the test session.
► Teaching method
  ▪ User interacts with the system to gain some expertise.
  ▪ A novice user is brought in, and the “expert” user is asked to explain to the novice user how to perform the task.
► Performance Measurement
  ▪ Obtaining quantitative data when participants perform a task.
  ▪ Minimize participant-tester interaction during the test as it might affect quantitative data.
  ▪ Careful design of number of participants to ensure enough statistical power.

Usability Evaluation Testing
► Question-asking Protocol
  ▪ Tester asked questions directly to participants.
  ▪ Participants answered in terms of their past experience in relation to the tested system.
► Remote Testing
  ▪ Tester are separated in time/space from users.
  ▪ Data are obtained from logs/records/networks.
► Retrospective Testing
  ▪ Tester & participants reviewed recorded session together, tester asked what was going on.
  ▪ Should be used together with other methods.