Basic Idea

- We covered the topics in bits and pieces earlier
  - E.g., models for evaluation, empirical evaluation

Basic Idea

- In this lecture, we'll try to get a comprehensive idea on
  - **Different evaluation techniques**
  - **Role of computing in evaluation**

Basic Idea

- Let's start with a basic question
  - **How do we evaluate a user-centric system?**

Basic Idea

- **Answer – by evaluating usability**
- What we evaluate?

Basic Idea

- The five measures
  - Learnability
  - Memorability
  - Efficiency
  - Error rate
  - Satisfaction

Basic Idea

- With computational user models (e.g., GOMS, Fitts' law), we can predict the *efficiency* measure

Basic Idea

- Some models are also capable of predicting *error rate* measure

Basic Idea

- Formal models can be used to measure certain quantities, which are linked to usability measures

Basic Idea

- Unfortunately, it is very difficult, if not impossible, to evaluate all the usability concerns with those models

Basic Idea

- Ex - **memorability**: how easy it is to remember the features of a design
  - There are attempts to use cognitive architectures
  - For complex systems, however, even those methods difficult

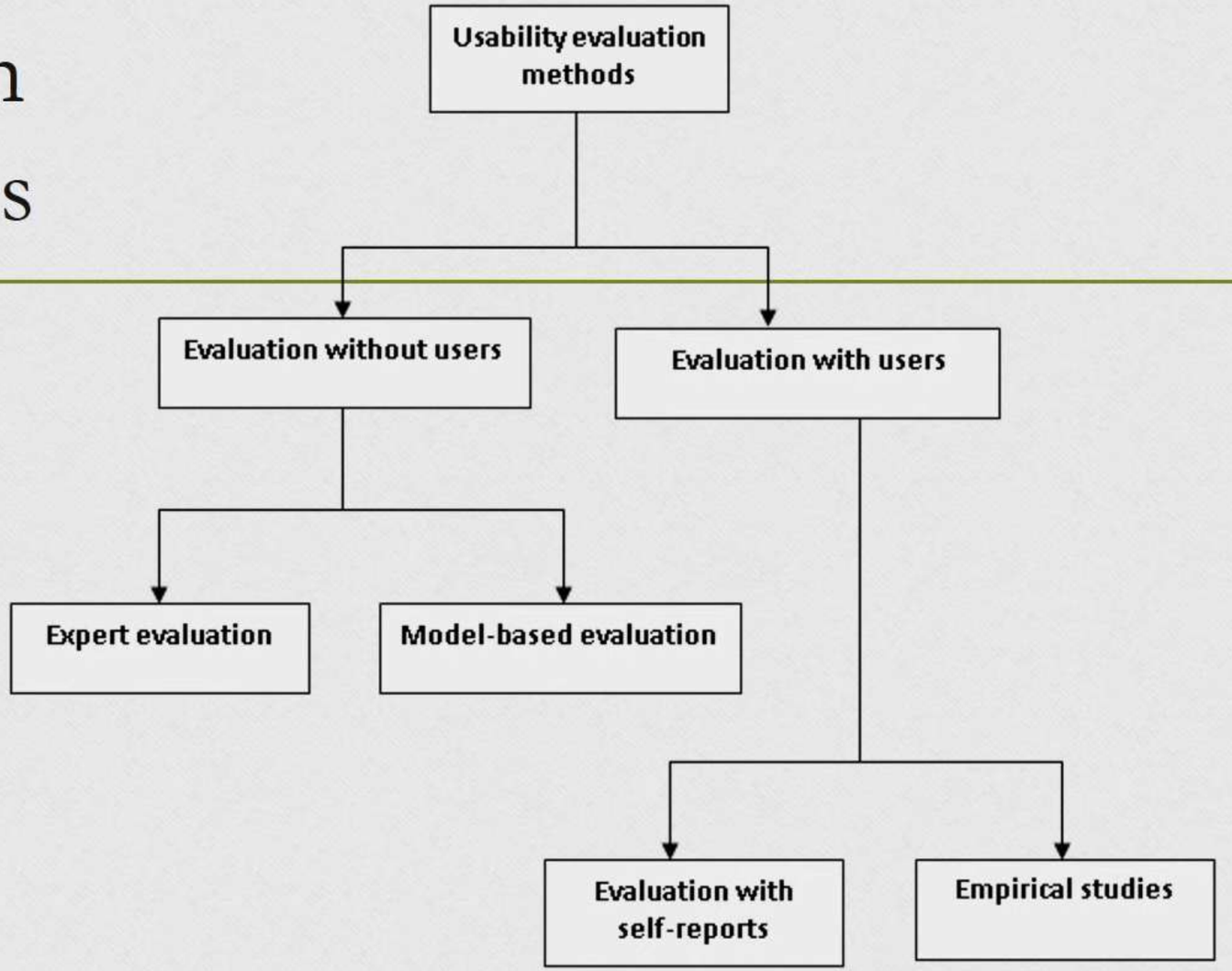
Basic Idea

- Similar situation with *learnability*
- Even more difficult is measure of *satisfaction*

Basic Idea

- Models can be used for some measures - cannot get comprehensive evaluation done with models **only**
- We should go for alternative evaluation techniques
- There are many such techniques

Evaluation Techniques



Evaluation Techniques

- Evaluation involve two activities
  - Collection of usage data
  - Analysis of data

Evaluation Techniques

- Data collection
  - Manual
  - Automatic (with technology aids)

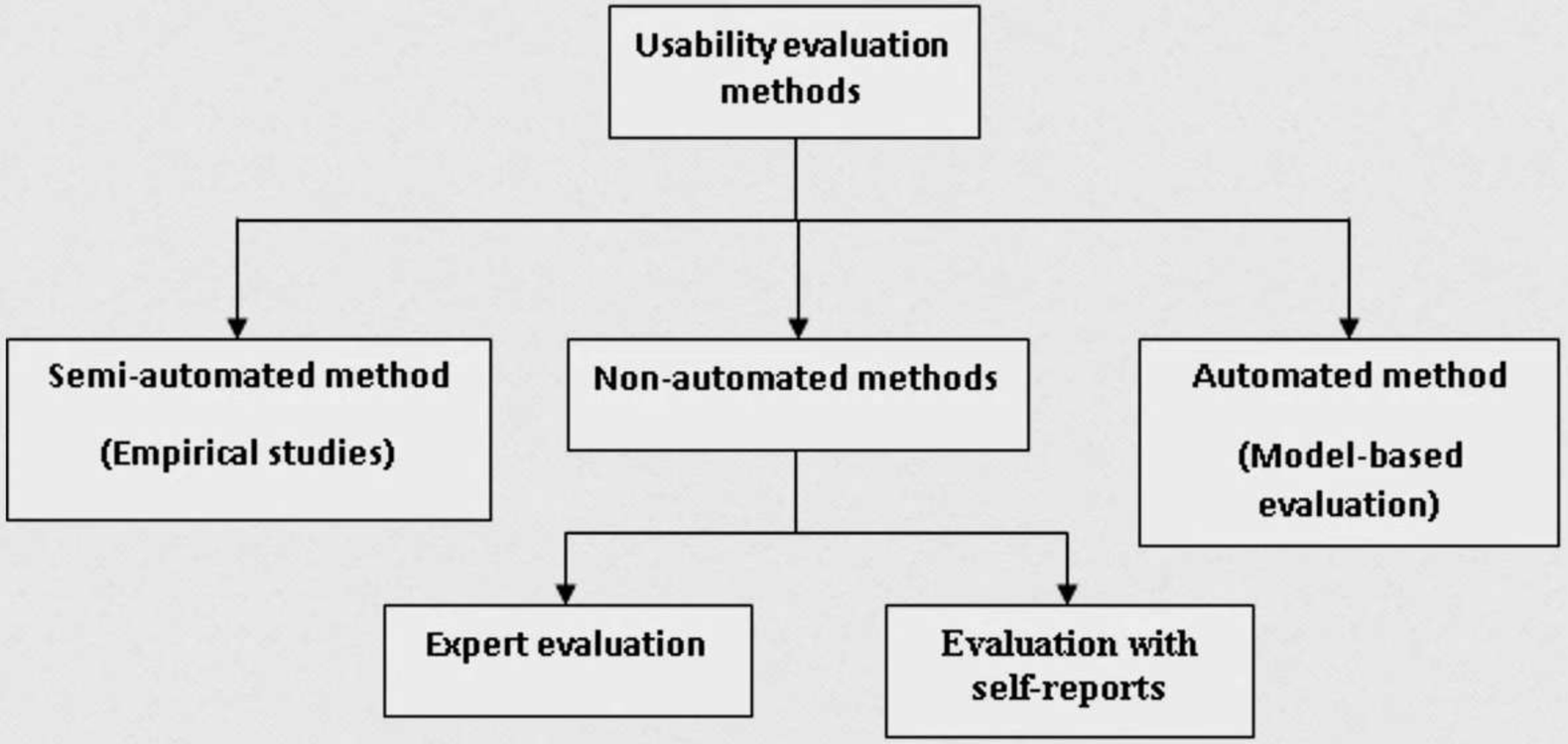
Evaluation Techniques

- Data analysis
  - With non-automated means (e.g., inspection-based approaches)
  - Automated tools

Evaluation Techniques

- Automated data analysis
  - **Semi-automated**: use of automation tools to process data (e.g., preparation of Tables, charts or statistical analysis) for further analysis
  - **Automated**: use of models to compare and conclude about the usability without any post-processing of data

Evaluation Techniques (2nd View)



Basic Idea

- Used for **quick and cheap** evaluation (typically done at early stages of design)
- We need TWO things

Basic Idea

- At least a **low-fidelity prototype**
- **An evaluation team**
  - May be the design team or may include other skilled designers
  - Optionally, few end users may also be included
  - Should have at least 3-5 members

Basic Idea

- Each team member evaluates individually and produces a report
  - Report contains a list of usability issues
- All these reports **combined** to produce a **final list**

Note

- We'll learn TWO of the expert evaluation methods

Cognitive Walkthrough

Basic Idea

- An **Usability Inspection Method**
- Requirements
  - At-least a low-fidelity prototype with support for several tasks (vertical prototype)
  - 3-5 member team (as mentioned earlier)

Example – Prototype Building

- Let us consider a simple calendar app. In our app, we show only the months in the first screen. Once the user selects a month, another screen appears showing the dates in that month. A user can select any date and add some note. What are the tasks that a user can do with this interface?

Example – Prototype Building

- Some of the tasks
  - Select a month
  - Select a day of a month
  - Add note
  - Get back to the month view from the day view

Example – Prototype Building

- To perform cognitive walkthough, we should make prototype that supports more than one of these tasks
- We can make simple paper prototypes for these tasks
  - Each prototype series of sketches depicting change of screen after each interaction

Additional Requirements – Task Description

- **Task descriptions** - specify scenario to perform tasks
- **Ex (calendar app):** You are planning to take a lecture on user-centric computing. You wish to schedule a class on the first Monday of the next month. The students are available only on the specific Monday. On all the other days, they do not have any free slot! You want to find out the date so that you can inform the students about the class.

Additional Requirements – Task Description

- Task - identify date on which first Monday of next month falls
- Interface-level tasks
  - Select the next month
  - Locate the first Monday
  - Note the date

Procedure

- The scenario is given to the evaluators
- They are asked to find out the date (by performing the interface-level tasks with the prototype)

Additional Requirements – Questions

- We also need to frame a set of questions (usability issues) beforehand
- Evaluators report on these issues while they perform the tasks

Additional Requirements – Questions

- We also need to frame a set of questions (usability issues) beforehand
- Evaluators report on these issues while they perform the tasks

Additional Requirements – Questions

- Example questions (calendar app)
  - Are you able to locate the month you are looking for easily?
  - Is the interaction required to change from the month view to the day view apparent?
  - Did you find it difficult to locate the first Monday?
  - Was the date clearly visible along with the day?
  - Did you try to go back to the month view? Was the mechanism to go back clearly visible?

Additional Requirements – Questions

- Purpose - identify problems user likely to face
- Evaluators not expected to answer only “yes” or “no”
  - Should give a detailed report on what they felt about the interfaces and interactions with respect to each question
- Feedbacks are analyzed to identify broader usability issues

Additional Requirements – Questions

- Is the effect of the action same as that of the goal of the user at that point?
- Will a user see the control for a particular action?
- Will a user see that the control produces the desired effect?
- Will a user select a different control instead?
- Will a user understand the feedback provided by the system to proceed correctly?
- What happens in case of an error?
- How a user, who is familiar with other systems that perform similar tasks, is going to react?

Basic Idea

- Cognitive walkthough useful in the early stages of design
- **Problem - method is scenario-based**
  - we evaluate w.r.t. specific usage scenarios
  - Complex systems likely to be numerous usage scenarios
  - Can't evaluate w.r.t. all

Basic Idea

- TWO reasons
  - First, we may not have that much time
  - Second (and more likely), we may not even be able to enumerate all possible usage scenarios

Basic Idea

- We can work with **representative use cases**
  - Not easy either

Basic Idea

- We may go for comprehensive evaluation of the whole system
  - We no longer need to create scenarios and ask evaluators to perform tasks
  - Instead, we can ask evaluators to **tick** on a checklist of features of the system as a whole

Basic Idea

- That is heuristic evaluation – **evaluate a system with a checklist**
  - Items in the checklist are called heuristics

Basic Idea

- Many such checklists available
  - Some are quite detailed and system specific
  - There are some that are more focused on broader principles of usability
- Latter category contains fewer heuristics

Basic Idea

- We shall discuss one from this latter category, which happens to be very popular as well – **the 10 heuristics by Nielsen** [Nielsen, 1994]

Nielsen's 10 Heuristics

Nielsen's ten heuristics	
Heuristic 1	Visibility of system status.
Heuristic 2	Match between system and the real world.
Heuristic 3	User control and freedom.
Heuristic 4	Consistency and standards.
Heuristic 5	Error prevention.
Heuristic 6	Recognition rather than recall.
Heuristic 7	Flexibility and efficiency of use.
Heuristic 8	Aesthetic and minimalist design.
Heuristic 9	Help users recognize, diagnose and recover from errors.
Heuristic 10	Help and documentation.

Setup & Procedure

- Low-fidelity prototypes - horizontal prototypes alright
  - Unlike the walkthrough method, which requires vertical prototypes

Setup & Procedure

- We also need a team of evaluators having similar team size (at least 3-5 members)
- Can be designers or other expert designers

Setup & Procedure

- Evaluation process slightly different
  - Task scenario not required - each evaluator checks design w.r.t. heuristics and report their findings
  - Reports are combined to determine heuristics that are violated

Book

- **Bhattacharya, S**. (July, 2019). Human-Computer Interaction: User-Centric Computing for Design, McGraw-Hill India
  - Print Edition: ISBN-13: 978-93-5316-804-9; ISBN-10: 93-5316-804-X
  - E-book Edition: ISBN-13: 978-93-5316-805-6; ISBN-10: 93-5316-805-8

Chapter 9, Sec 9.1-9.2