IPM 11/12 – T1.6
Discount Evaluation Methods

Licenciatura em Ciência de Computadores

Miguel Tavares Coimbra

Acknowledgements: Most of this course is based on the excellent course offered by Prof. Kellogg Booth at the British Columbia University, Vancouver, Canada. Please acknowledge the original source when reusing these slides for academic purposes.

U.PORTO FC

Summary

- Discount Evaluation Methods
- Cognitive Walkthrough
- Heuristic Evaluation

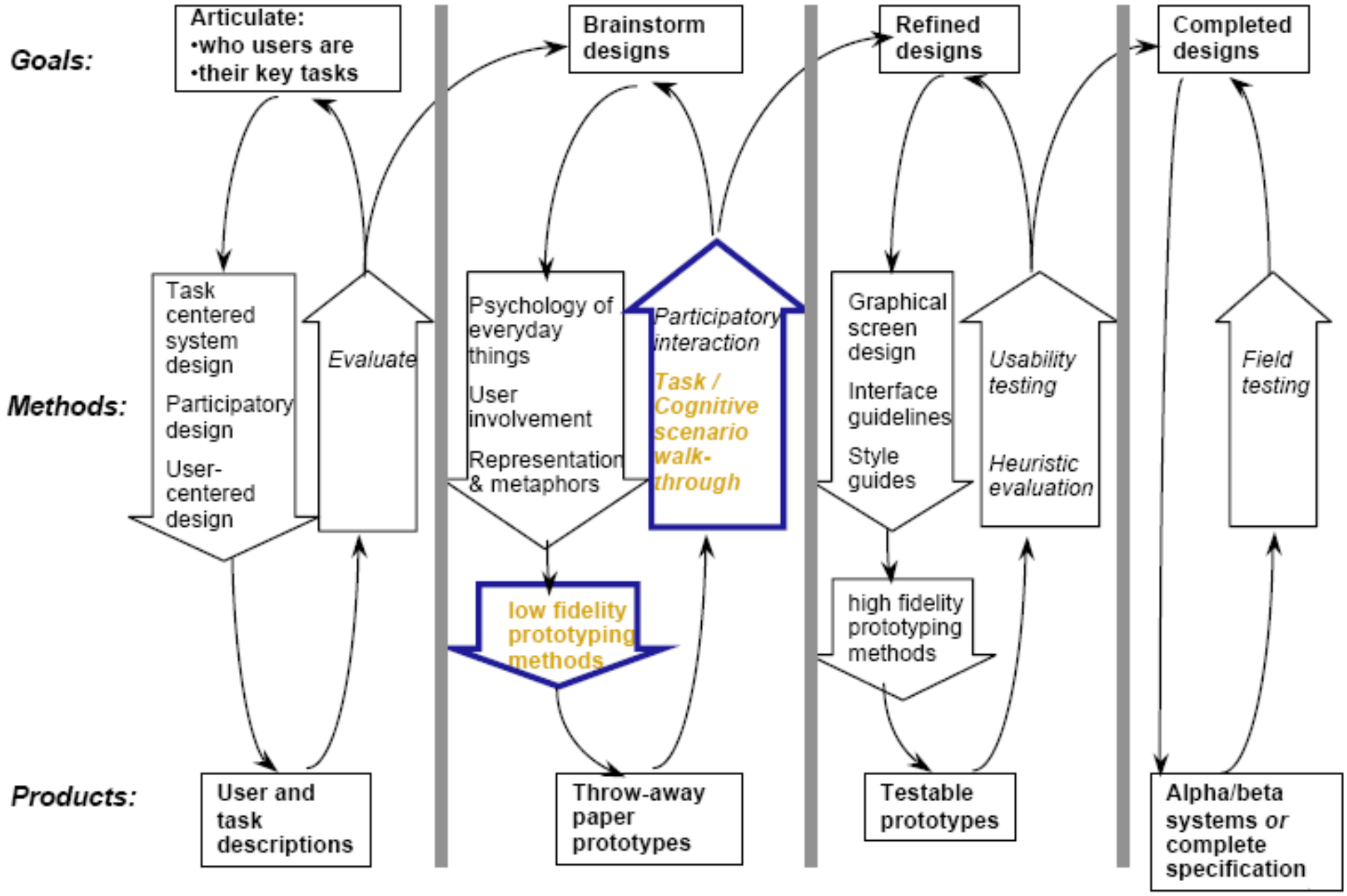
Discount usability engineering

- Cheap (thus 'discount')
  - No special labs or equipment needed
  - Doesn't need to involve users *directly*
  - the more careful (and informed by users) you are, the better it gets
- Fast
  - On order of 1 day to apply
  - Standard usability testing may take a week
- Easy to use
  - Can be taught in 2-4 hours

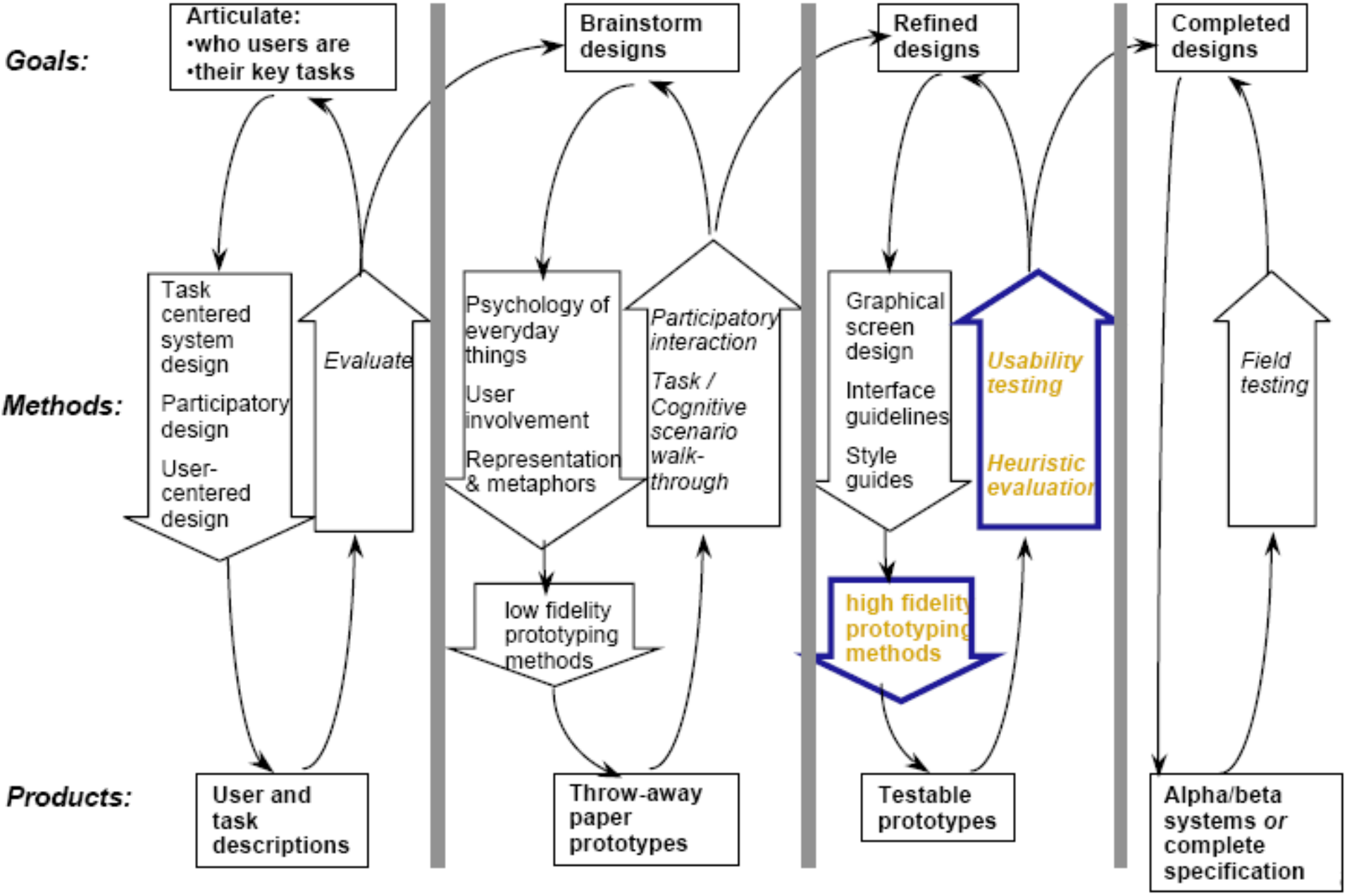
Types of discount methods

- Cognitive walkthrough: **"mental model"**
  - Assesses "exploratory learning stage" (new users)
  - *What mental model does the system image facilitate?*
  - Done by non-experts and/or domain experts
- Heuristic evaluation: **"fine tune"**
  - Targets broader use range (including expert)
  - Fine-tunes the interface
  - HCI professionals apply a list of heuristics while simulating task execution

Interface Design and Usability Engineering



Interface Design and Usability Engineering



Cognitive walkthrough
Exploratory learning

- **What for**: assessing how well a **new user** will be able to figure out the interface
- **Not for**: assessing performance at highly skilled, frequently performed tasks; or finding radically new approaches
- **Additional advantages**: helps work out task sequence models through observation
- **Disadvantages**: limited utility for frequent-use interfaces, narrow focus, relatively time consuming & laborious (compared to HE)

Cognitive walkthrough

- Possible outputs:
  - Loci & sources of confusion, errors, dead ends
  - Estimates of success rates, error recovery; performance speed less evident
  - Helps to figure out what activity sequences could or should be
- What's required: complete interface description
  - (e.g., a paper prototype)
- Who does it:
  - anyone – different benefits will accrue from using design team members, naïve users or expert outside analysts. **More distance = better!**
- *Alternate spec for paper prototype:*
  - Must accommodate a cognitive walkthrough

How? Roughly:

- Start with a scenario (a **design-specific task)**
- Ask these questions at each step as relevant:
  - Q1: Will the correct action be evident?
  - Q2: Will the user recognize the correct action?
  - Q3: Will the user interpret the result correctly?
  - Q4: Will the user be able to progress towards goal?

discount method #2: heuristic evaluation

- **What for:** identifying (listing & describing) problems with existing prototypes (any kind of interface)
- **Not for:** coming up with radically new solutions
- **Additional advantages**: contributes valuable insights from objective observers
- **Disadvantages:**
  - Reinforces existing design - better solutions might exist
  - Not very repeatable

Heuristic evaluation

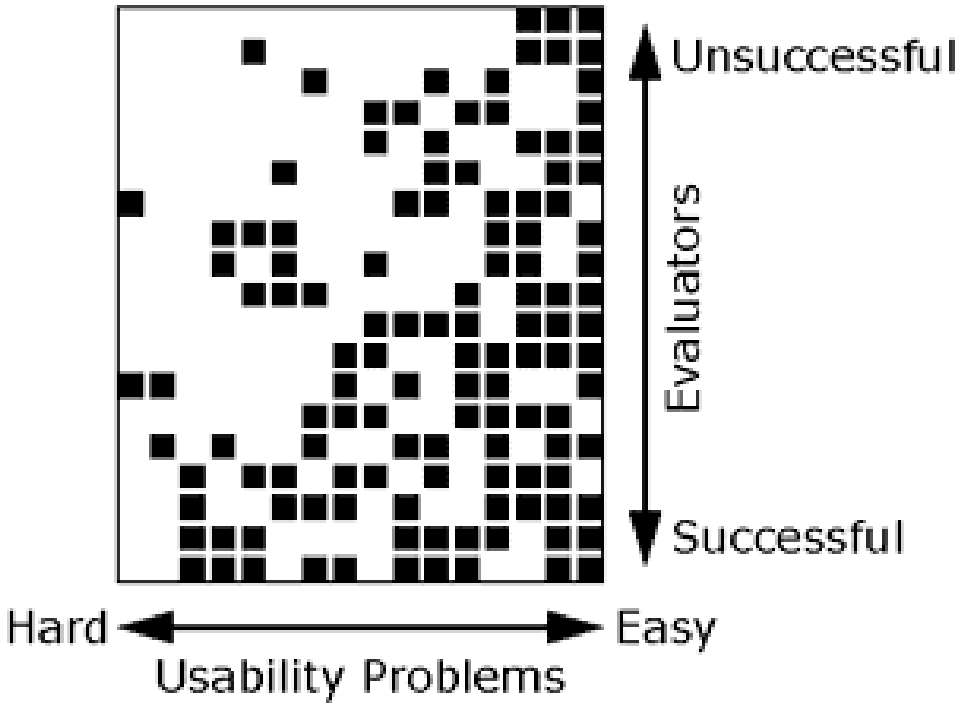
- What's required:
  - A good model of the proposed interface (e.g., at least a paper prototype)
  - A list of design heuristics to be applied
  - A scenario (task example + design prototype)
- Who does it:
  - Team of 3 to 5 experienced, objective people ("**experts")** who **aren't on the design team.**
- General idea:
  - Independently check compliance with usability principles ("heuristics")

step 1: each evaluator works with interface *alone* (different evaluators will find different problems)

step 2: evaluators aggregate findings afterwards

Why multiple evaluators?

- Every evaluator doesn't find every problem
- Proficient evaluators find both easy & hard (subtle) ones



One list of heuristics (Nielson, '93)

- H2-1: visibility of system status
- H2-2: match between system & the real world
- H2-3: user control & freedom
- H2-4: consistency and standards
- H2-5: error prevention
- H2-6: recognition rather than recall
- H2-7: flexibility and efficiency of use
- H2-8: aesthetic and minimalist design
- H2-9: help users recognize, diagnose & recover f/ errors
- H2-10: help and documentation

Step 1: Individual evaluation

- At least two passes for each evaluator
  - First to get feel for flow and scope of system
  - Second to focus on specific elements
- Each evaluator produces list of problems
  - Explain problem w/reference to heuristic or other info
  - Be specific and **list each problem separately**
  - Assign rating of **severity** to each violation
- **Tips:**
  - Be respectful but critical
  - Let your client decide whether to ignore a problem
  - Look especially for what's *not there*

Severity ratings

- Each violation is assigned a **severity rating**
- Combination of:
  - Frequency
  - Impact
  - Persistence (one time or repeating)
- Used to:
  - Allocate resources to fix problems
  - Estimate need for more usability efforts
- Done independently by all evaluators

  note: in SJWM, severity is set later in process; extent not used

Severity & extent scales

- One severity scale (others possible):
  - 0 - don't agree that this is a usability problem
  - 1 - cosmetic problem
  - 2 - minor usability problem
  - 3 - major usability problem; important to fix
  - 4 - usability catastrophe; imperative to fix
- One extent scale:
  - 1 = single case
  - 2 = several places
  - 3 = widespread

Step 2: aggregating results & making recommendations

- **Evaluation team** meets and compares results
- Through discussion and consensus, each violation is documented and categorized in terms of severity, extent
- Violations are ordered in terms of severity

→ **combined report goes back to design team.**

Summary: how to perform Heuristic Evaluation

1. Design team supplies scenarios, prototype; need 3-5 evaluators
2. Each evaluator **independently** produces list of justified, rated problems by stepping through interface and applying heuristics at each point
   … use heuristics list & severity rating convention
3. Team meets and compiles report that organizes and categorizes problems

Summary: heuristic evaluation

- Advantages
  - The “minimalist” approach
    - General guidelines can correct for majority of usability problems
    - Easily remembered, easily applied with modest effort
    - **“black box”: systematic technique that is reproducible with care.**
  - *Discount usability engineering*
    - Cheap and fast way to inspect a system
    - Can be done by usability experts and end users
- Problems:
  - Principles must be applied intuitively and carefully
    - Can’t be treated as a simple checklist
    - Subtleties involved in their use
  - Doesn’t necessarily predict users/customers’ overall satisfaction
  - May not have same “credibility” as user test data
    - A solution: include design team & developers in usability evaluation

Summary: heuristic eval, cont.

- Research result:
  - 4-5 evaluators usually able to identify 75% of usability problems
  - User testing and usability inspection have a large degree of non-overlap in the usability problems they find (i.e., it pays to do both)
- Cost-benefit:
  - Usability engineering activities often expensive / slow; but some can be quick / cheap, and still produce useful results
  - Usability inspection turns less on what is “correct” than on what can be done within development constraints
  - Ultimate trade-off may be between doing *no usability assessment* and doing *some kind*

Resources

1. Kellogg S. Booth, Introduction to HCI Methods, University of British Columbia, Canada http://www.ugrad.cs.ubc.ca/~cs344/current-term/