

# Praktikum Entwicklung von Mediensystemen mit iOS

SS 2012

Prof. Dr. Michael Rohs  
michael.rohs@ifi.lmu.de  
MHCI Lab, LMU München

# Today

- Evaluation

# Timeline

#	Date	Topic
	19.4.	Introduction & Brainstorming future mobile concepts
1	3.5.	Video watching, Introduction to iOS
	10.5.	no class (CHI Konferenz)
	17.5.	no class (Christi Himmelfahrt)
2	24.5.	More on iOS
3	31.5.	Concept finalization, paper prototyping
	7.6.	no class (Fronleichnam)
4	14.6.	Paper prototyping test, start of software prototype
5	21.6.	
6	28.6.	Think aloud study of software prototype
7	5.7.	
8	12.7.	Completion of software prototype
9	19.7.	Final presentation

# AUFGABEN

# Exercise 3

- Ziele

- Die Location API kennenlernen
- Verstehen fremden Programmcodes
- Animationen verwenden

## Übungsblatt 3

### Ziele

- Die Location API kennenlernen
- Verstehen fremden Programmcodes
- Animationen verwenden

### Aufgabe 1

Auf der Webseite der Vorlesung findet sich eine an iOS5 angepasste Version der „LocateMe“ [2] Beispielanwendung. Laden Sie die iOS5-Version von der Webseite herunter und öffnen Sie sie in Xcode. Stellen Sie das „Active Scheme“ um auf „iPhone 5.x Simulator“. Der iOS5-Simulator erlaubt die Eingabe beliebiger GPS-Koordinaten (im Simulator Debug | Location | Custom Location...).

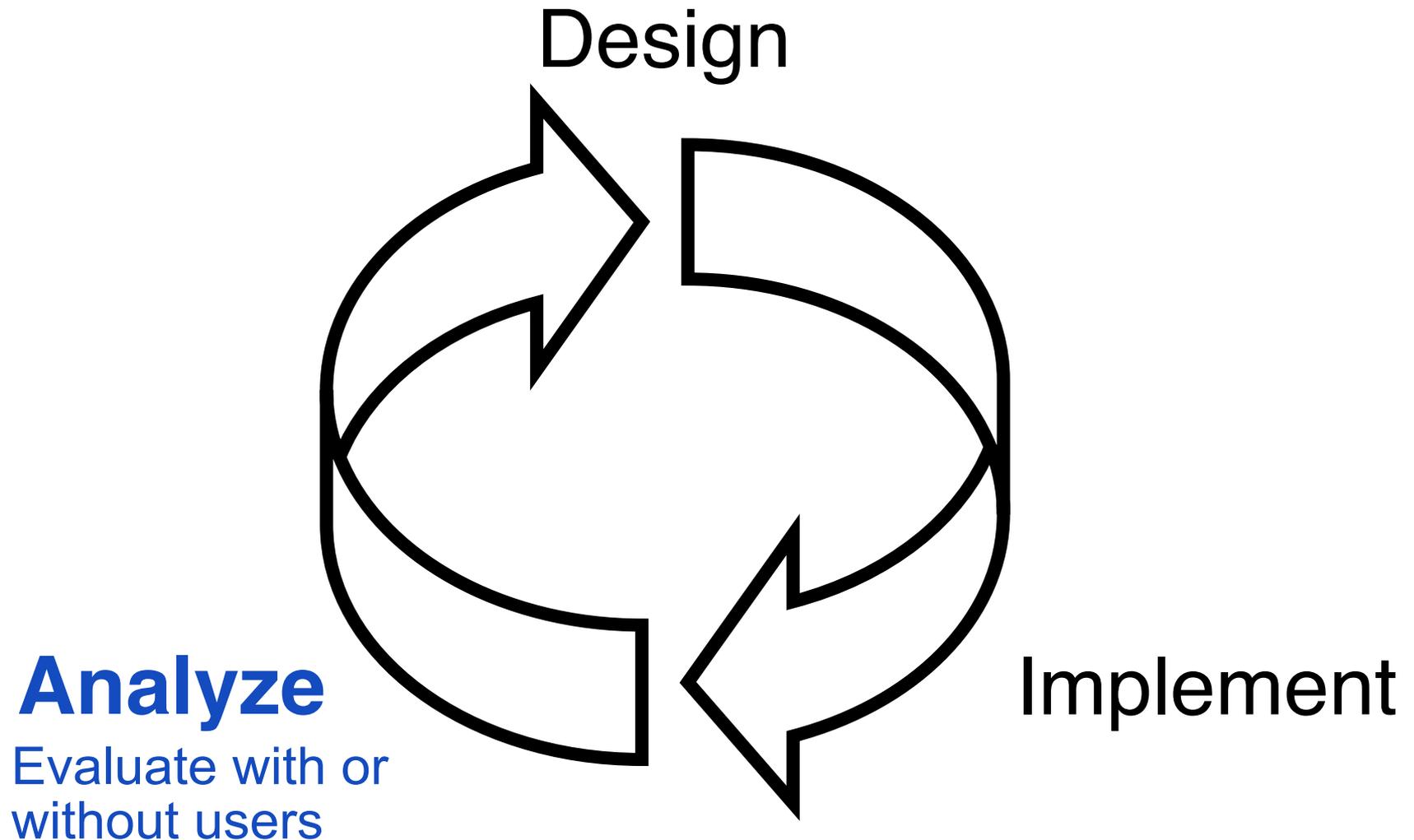
- Analysieren Sie zunächst das Storyboard. Vergleichen Sie die Struktur (besonders des Tab Bar Controllers) mit der Struktur Ihrer Tab-Bar-Anwendung aus der letzten Übung. Worin bestehen die Unterschiede und was wird durch die Struktur in „LocateMe“ zusätzlich ermöglicht?
- Begründen Sie, warum es nicht notwendig ist, dass die Start-Buttons in den Szenen „Get Location“ und „Track Location“ mit IBAction-Methoden verbunden sind. Erklären Sie grob die Schritte, die das Programm nach Drücken des Start-Buttons in der „Get Location“ Szene ausführt.
- Machen Sie sich mit der Funktionsweise des „Picker Controls“ in der Szene „Setup View Controller“ vertraut und erklären Sie die grundlegende Funktionsweise (Woher kommen die Daten, die im Picker zu sehen sind? Was bewirkt die Interaktion des Benutzers?). Sie müssen nicht alle Funktionen des „Picker Controls“ erklären, sondern nur die, die in der Controller-Klasse (SetupViewController.{h,m}) verwendet werden. Wozu wird das Dictionary „setupInfo“ in der Klasse SetupViewController verwendet? Wie werden die Daten vom SetupViewController zurück zum Get- bzw. TrackLocationViewController transferiert? Hinweis: [obj respondsToSelector:@selector(method)] überprüft, ob das Objekt obj die Methode method implementiert.
- Wozu dient die Klasse LocationDetailViewController? Wieso taucht sie nicht im Storyboard auf? Wann wird die Klasse instanziiert? Wie viele Instanzen der Klasse gibt es zur Laufzeit maximal?
- „CLLocation (Strings)“ ist ein Beispiel für eine Objective-C „Category“. Damit lässt sich die Funktionalität existierender Klassen durch Hinzufügen neuer Methoden erweitern (in diesem Fall CLLocation), ohne dass man (wie bei Vererbung) die existierenden Methoden der Klasse ändert. Categories können nur Methoden, aber keine Klassen- oder Instanzvariablen hinzufügen. Beschreiben Sie die Funktionalität der Category „CLLocation (Strings)“ und die Bedeutung der Localized Strings.

# Aufgabe

- Erstellen Sie einen Papier-Prototyp, der die einzelnen Screens/UI-Zustände und Übergänge dazwischen darstellt.
- Führen Sie einen Benutzer-Test mit dem Papier-Prototyp mit 3 Benutzern (nicht aus dem PEM-Praktikum) durch
  - Typische Aufgabe festlegen
  - Benutzer bitten, diese Aufgabe mit dem Papier-Prototyp durchzuführen
  - Protokollieren, Probleme finden
  - Papier-Prototyp verbessern
- Schreiben Sie die Erfahrungen beim Test auf
  - Was ließ sich gut testen, was war problematisch

# EVALUATION

# DIA Cycle: When to evaluate?



# User Interface Guidelines

- Concrete guidelines for look-and-feel and behavior
  - Visual appearance, e.g., icon design
  - Purpose of user interface elements
  - Layout of user interface elements
  - Behavior, conventions of system features
- Android User Interface Guidelines
  - [http://developer.android.com/guide/practices/ui\\_guidelines/index.html](http://developer.android.com/guide/practices/ui_guidelines/index.html)
- iOS Human Interface Guidelines
  - <http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/MobileHIG.pdf>
  - Aesthetic integrity, consistency, direct manipulation, feedback, metaphors, user control, ...

# Four Fundamental Concepts (Donald Norman)



- Affordances & visibility
  - Affordances
    - [http://www.jnd.org/dn.mss/affordances\\_and\\_design.html](http://www.jnd.org/dn.mss/affordances_and_design.html)
  - Can the user tell the state of the system and the alternatives for action by looking at the system?
- Conceptual models
  - Is the user able to predict how actions affect the system?
- Natural mapping
  - Is it possible to determine the relationships between actions and results, between controls and effects?
- Feedback
  - Does the user receive full and continuous feedback about the results of actions?

# Heuristic Evaluation

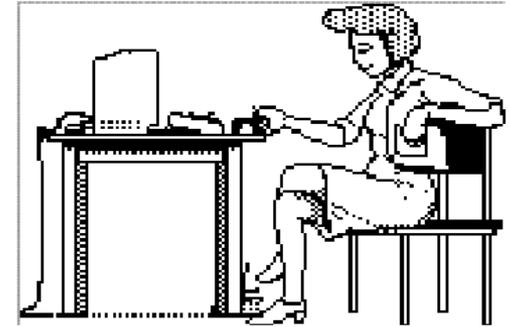
- Choose usability heuristics
  - (general usability principles, e.g., Nielsen's 10 Usability Principles)
- Step through tasks and check whether guidelines are followed
- Severity rating for each problem (Nielsen)
  - 0 = I don't agree this is a problem at all
  - 1 = cosmetic problem
  - 2 = minor usability problem, low priority to fix
  - 3 = major usability problem, high priority to fix
  - 4 = usability catastrophe, imperative to fix before release
- + Quick and cheap
- Subjective (have several independent evaluators)
  - See also: [www.useit.com/papers/heuristic](http://www.useit.com/papers/heuristic)

# 10 Usability Principles (Jakob Nielsen)

1. Keep the interface simple!
2. Speak the user's language!
3. Minimize the user's memory load!
4. Be consistent and predictable!
5. Provide feedback!
6. Design clear exits and closed dialogs!
7. Offer shortcuts for experts!
8. Help to recover from errors, offer Undo!
9. Prevent errors!
10. Include help and documentation!



# Silent Observation



Source: Saul Greenberg

- Designer watches user in lab or in natural environment while working on one of the tasks
- No communication during observation
- + Helps discover big problems
- No understanding of decision process (that may be wrong) or user's mental model, opinions, or feelings

# Think Aloud



Source: Saul Greenberg

- As Silent Observation, but user is asked to say aloud
  - What he thinks is happening (state)
  - What he is trying to achieve (goals)
  - Why he is doing something specific (actions)
- Most common method in industry
- + Good to get some insight into user's thinking, but:
  - Talking is hard while focusing on a task
  - Feels weird for most users to talk aloud
  - Conscious talking can change behavior

# Evaluation in the Mobile Context

- Context of use needs to be taken into account
  - Factors: User, activity, device, environment
- Usage “on the move”
  - Physically moving: walking, driving a car, traveling as a passenger
  - Being in different places: away from office environment or home
- Difficult to collect data in the field
  - Recording data
  - Controlling experimental conditions



# AUFGABE

# Aufgabe

- Ziel der Aufgabe: Test des Paper-Prototyps ohne Benutzer mittels heuristischer Evaluation
- Heuristiken sind allgemeine Regeln, die beim Entwurf der Benutzungsschnittstelle beachtet werden sollten.
  - Verwenden Sie Niensens aktualisierte Liste von “Ten Usability Heuristics” als Basis (Folien, sowie [www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)).
  - Fügen Sie dieser Liste basierend auf der bisherigen Konzeptentwicklung mindestens eine weitere Regel hinzu, die spezifisch für Ihr Projekt sind.

# Aufgabe

- Vorgehen
  - Einigen Sie sich im Team auf Szenario und Test-Aufgabe
  - Jedes Team-Mitglied führt unabhängig (!) eine heuristische Evaluation durch
  - Bewerten Sie die Wichtigkeit der gefundenen Usability-Probleme
  - Tragen Sie die Resultate zusammen, vergleichen Sie die Resultate und ordnen Sie nach Wichtigkeit
  - Ändern Sie den Prototyp, um die Probleme zu lösen

# Aufgabe

- Lesen Sie die folgenden Quellen als Hintergrundinformation:
  - Heuristic evaluation:  
[www.useit.com/papers/heuristic](http://www.useit.com/papers/heuristic)
  - How to conduct HE:  
[www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)
  - On severity ratings:  
[www.useit.com/papers/heuristic/severityrating.html](http://www.useit.com/papers/heuristic/severityrating.html)

# Aufgabe

- Beim nächsten Treffen soll jede Gruppe folgendes vortragen:
  - Liste zusätzlicher Heuristiken (2-4)
  - Gefundene Usability-Probleme (Text / Bilder / mündlicher Vortrag)
  - Bewertung der Wichtigkeit (Severity Rating) jedes Problems
  - Beschreibung der Änderungen am Prototyp (Text / Bilder / mündlicher Vortrag)
  - Kurze mündliche Beschreibung der Erfahrungen mit heuristischer Evaluation (Bewertung, Aufwand, etc.)
- Arbeiten Sie weiter an Ihrem Software-Prototyp. Zeigen Sie beim nächsten Treffen kurz den aktuellen Stand.