Arbeitsgruppe Software Engineering,
Institut für Informatik

Freie Universität Berlin

Abschlussvortrag der Bachelorarbeit:

Muster im User Interface Reengineering in ein deklaratives User Interface Framework

Adrian Meister

10.11.2022

Betreuet von Prof. Dr. Lutz Prechelt

Freie Universität Berlin

Gliederung

- Applikation und Domäne
- Motivation
- Ziele
- Stand der Kunst
- Ansatz
- Umsetzung und Ergebnisse
- Fazit und Ausblick

Freie Universität Berlin

Applikation und Domäne

Applikation

- Unleashed App (iOS) entwickelt bei Foolography GmbH
- Steuerung des gleichnamigen Hardwaremoduls
- Dient zur Kontrolle von Kameras

Freie Universität Berlin

Applikation und Domäne

Domäne

- Benutzeroberfläche / User Interface (kurz UI)
- View eine UI-Komponente / UI-Datentyp im Programmcode
- Imperative und deklarative UI-Programmierung

Freie Universität Berlin

Motivation

- Mehrere gescheiterte Refactoring-Anläufe
- Lebenszyklus der imperative UI-Basis erreicht
- Umstieg auf zukunftssichere deklarative UI-Basis

Freie Universität Berlin

Ziele

User Interface

- Deckung der Funktionalität
- Ausreichende Dokumentation
- Eigenständiges Swift Package
- UI-Tests (gelockert - Testfälle zur Laufzeit)

Freie Universität Berlin

Ziele

Einschränkungen bei Implementierung

- Keine unmittelbare Integration in die App
- Nur eine selektierte Teilmenge von Views (keine Navigation)
- Keine stilistischen Anteile

Freie Universität Berlin

Ziele

Muster

- Setzt Volumenarbeit voraus
- Gibt es überhaupt (neue) Muster?
- Für den späteren Einsatz in der Praxis
- Einfache Darstellung

Freie Universität Berlin

Stand der Kunst

UIKit

- Imperatives UI-Framework von Apple
- Entwickelt in Objective-C

Freie Universität Berlin

Stand der Kunst

SwiftUI

- Erscheint magisch und ist sehr komprimiert
- Result Builder (früher Function Builder)
  - Baut die deklarative Syntax
- Property Wrapper
  - Arbeiten und Manipulation von Programmzuständen

Freie Universität Berlin

Stand der Kunst

Model-View-ViewModel als Stützmittel

- Sehr populär für App-Entwicklung
- Dient der Trennung des Models von dem View
- ABER:
  - Keine konkrete Empfehlung von Apple
  - Es gibt Argumente gegen den Einsatz in SwiftUI

Freie Universität Berlin

Ansatz

Visuelle Analyse

- Betrachtung der App
- Auflistung von optisch semantischen vermuteten Views
- Verteilung von Schwierigkeitsgraden:
  - Schwer (2): gesondertes Scrollverhalten
  - Mittelschwer (3): komplexe Animationen, gesonderte Selektion-Fähigkeit
  - Leicht (13): nur Darstellung des Zustands, keine Animationen

Freie Universität Berlin

Ansatz

Funktionsumfangsanalyse

- Aus der Sicht eines neuen Entwicklers
- View-Debugger als erstes Hilfsmittel
- Suche nach Schlüsselwörtern
- Erstellung einer imperativen Pseudocode-Darstellung basierend auf MVVM

Freie Universität Berlin

Umsetzung und Ergebnisse

Implementierung

- Jede View erhält eine Preview + UI-Tests
- Großzügige Einschätzung der Zeit

Einfache Views

- Kein MVVM notwendig gewesen
- Verzicht auf Animationen

Freie Universität Berlin

Umsetzung und Ergebnisse

Implementierung / Schwere Views: VPicker und HPicker

- Zeitliche Einschätzung 6-8 Stunden (falsch)
- Zwei Ansätze:
  - ScrollView mit LazyStack (SwiftUI) —> UIScrollView (UIKit)
  - List (nur vertikal) (SwiftUI) —> UITableView (UIKit)
- Eigene Introspecting Entwicklung
- UIScrollView nutzt das Delegate Entwurfsmuster

Freie Universität Berlin

Umsetzung und Ergebnisse

Implementierung / Schwere Views: VPicker und HPicker

- Optionen für mögliche Lösungen:
  1) Proxy Delegation
  2) Überwachung der Swipe-Geste
  3) Implementierung zur Laufzeit mit Objective-C Runtime

Freie Universität Berlin

Umsetzung und Ergebnisse

Implementierung / Schwere Views: VPicker und HPicker

- Schwierigkeiten:
  1) Reine Injizierung von fehlenden Delegate-Methoden (falsch)
  -> ISA-Swizzling
  2) Unerwartete Fehlfunktion
  3) super-Aufruf
  4) Namensgebung
  5) Initiale Selektion

Freie Universität Berlin

Umsetzung und Ergebnisse

Muster und Erkenntnisse

1) Divide And Conquer
- Einschätzung der Schwierigkeitsgrade
- Aufteilung in kleinere Teilaufgaben
- Klarer Fokus und reduzierte Komplexität

2) Build Prototypes
- Deklarative Implementierung ist nicht immer eindeutig
- Wegwerfcode ist hilfreich

Freie Universität Berlin

Umsetzung und Ergebnisse

Muster und Erkenntnisse

3) Kein MVVM notwendig

4) Introspecting, Swizzling
- **Falls keine native Lösung, dann möglicherweise Introspecting**
- **Swizzling als letzten Ausweg (falls überhaupt möglich)**

Freie Universität Berlin

Fazit und Ausblick

Fazit

- Wiederholte Muster erkennbar
- Trotz massiven Problemen ein Erfolg
- Komplexität tendenziell korrekt einstuft, nicht aber den Zeitfaktor
- Swift-Package Teilweise in proof-of-concept Zustand

Ausblick

- Das Swift-Package aufräumen, fertigstellen und dokumentieren
- Reengineering von weiteren UI-Komponenten
- Anwendung der Muster und Techniken in der Praxis

Freie Universität Berlin

Quellen

Internetquellen

- Foolography GmbH: https://www.foolography.com/
- Deklarative Programmierung: https://de.wikipedia.org/wiki/Deklarative_ Programmierung
- Calling Super at Runtime in Swift: https://steipete.com/posts/calling- super-at-runtime/
- Delegation: https://docs.swift.org/swift-book/LanguageGuide/ Protocols.html#ID276
- Every question and answer from WWDC 22’s SwiftUI digital lounge! - Architecture: https://midnight-beanie-ccb.notion.site/swiftui-lounge-wwdc22-e20094b91f074398ba395c3fa245e63d
- Introspect for SwiftUI https://github.com/siteline/SwiftUI-Introspect

Freie Universität Berlin

Quellen

Internetquellen

- Kaan Enes KAPICI, Hrsg. Procedural(Imperative) UI vs Declerative UI and Approach on the Android Side. 10. Apr. 2021: https://kaaneneskpc.medium.com/ procedural-imperative-ui-vs-declerative-ui-and-approach-on-the-android-side-1725275c53f4
- Model View ViewModel: https://de.wikipedia.org/wiki/Model_View_ViewModel
- Unleashed - Camera Remote (AppStore): https://apps.apple.com/de/app/unleashed-camera-remote/id1361146202

Literaturquellen

- James O. Complin und Neil B. Harrison. Organiszational Patterns of Agile Software Development. Illustrated edition (15 July 2004). Pearson Education, 2004. isbn: 978-0131467408.

Freie Universität Berlin

Vielen Dank für Ihre Aufmerksamkeit!