

**Macoun**



# Storyboards: Eine Fallstudie

Martin Winter

# Ablauf

- Ziele des Vortrags
- Kurze Demo der Beispiel-App
- Einführung in Storyboards
- Grundkonzepte (mit Live-Beispielen)
- Vor- und Nachteile
- Tips und Muster

# Ziele des Vortrags

- Interesse wecken
- Erfahrungen weitergeben

# Die Beispiel-App

- »BasisBibel«, Deutsche Bibelgesellschaft, Stuttgart
- Buch, Website, App
- Universal (iPhone + iPad)
- Komplexität durch zusätzliche Text- und Medieninhalte

Demo I

# Einführung in Storyboards

- Film
- XIB: eine View (meistens)
- Storyboard: mehrere Views (bzw. View Controller) + Übergänge
- Können gemischt werden
- Nur iOS
- Edge Cases 22, »Nib and Xibs and Storyboards, O My«:  
ähnlich Resources unter Classic Mac OS

# Einführung in Storyboards

- [iOS 6 Documentation > User Experience > Windows & Views > View Controller Programming Guide for iOS > Using View Controllers in Your App > Working with View Controllers in Storyboards](#)
- [iOS 6 Documentation > Cocoa Touch Layer > UIKit > Converting to Storyboards Release Notes](#)
- [WWDC 2012, Session 407: »Adopting Storyboards«](#)
- [iDeveloper Live, Ep. 43: »Didn't See That Mountain Coming!«](#)



# Grundkonzepte

- Scene
- Segue
  - Relationship (z. B. UINavigationController—Root)
  - Transition (z. B. Push, Modal, Popover)
  - Neu in iOS 6: Embed, Unwind
  - `-[UIViewController prepareForSegue:sender:]`
  - Neu in iOS 6: `-shouldPerformSegueWithIdentifier:sender:`

# Demo 2

# Grundkonzepte

- Komplexe Abläufe möglich
  - Weichen
  - Schleifen
  - Mehrere Ein-/Ausgänge
- Achtung: Dekomposition! (dazu später mehr)

# Demo 3

# Table Views

- Dynamic Prototypes
- Static Cells
  - Trotzdem dynamischer Inhalt: Outlets
- Auch für Rapid Prototyping geeignet (s.Vorteile)

# Table Views

- Mit Automatic Cell Loading kombinierbar

```
UITableViewCell *cell =  
    [tableView dequeueReusableCellWithIdentifier:CellIdentifier];  
  
/* Alles ab hier entfällt mit Automatic Cell Loading! */  
  
if (!cell)  
{  
    cell = [[UITableViewCell alloc]  
            initWithStyle:UITableViewCellStyleDefault  
            reuseIdentifier:CellIdentifier];  
}
```

# Demo 4

# XIBs

- Mit Storyboards kombinierbar
- Brauchen keinen View Controller
- Beispiel: komplexe Subklasse von UITableViewCell



# Demo 5

# Storyboards und Code

- UIStoryboard aus Bundle laden

```
UINavigationController *storyboard =  
    [UINavigationController storyboardWithName:storyboardName  
                                     bundle:nil];
```

# Storyboards und Code

- UIViewController instantiieren

```
id initialViewController =  
    [storyboard instantiateInitialViewController];
```

```
id otherViewController =  
    [storyboard instantiateViewControllerWithIdentifier:@"foo"];
```

# Storyboards und Code

- UIStoryboardSegue manuell ausführen

```
[self performSegueWithIdentifier:@"bar"  
                                sender:self];
```

# Storyboards und Code

- Zugriff auf Source und Destination

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue
    sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"baz"])
    {
        MyViewController *source =
            (MyViewController *)segue.sourceViewController;
        MyViewController *destination =
            (MyViewController *)segue.destinationViewController;
        ...
    }
}
```

# Storyboards und Code

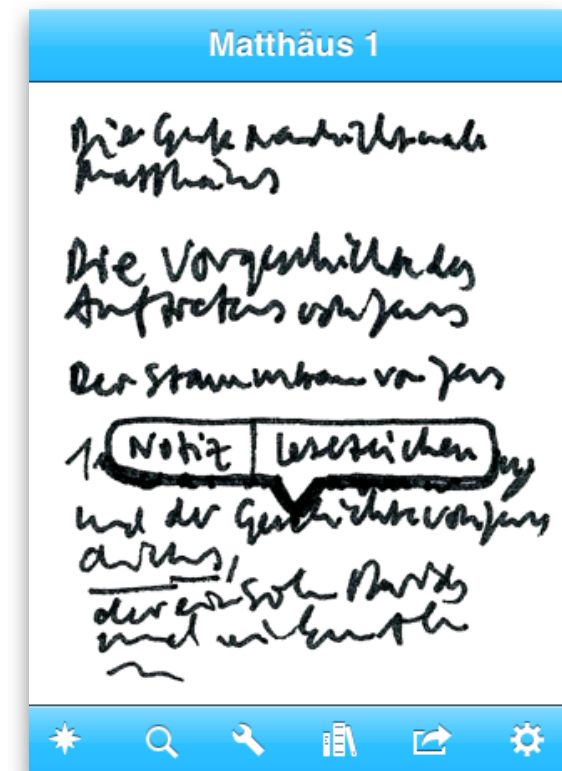
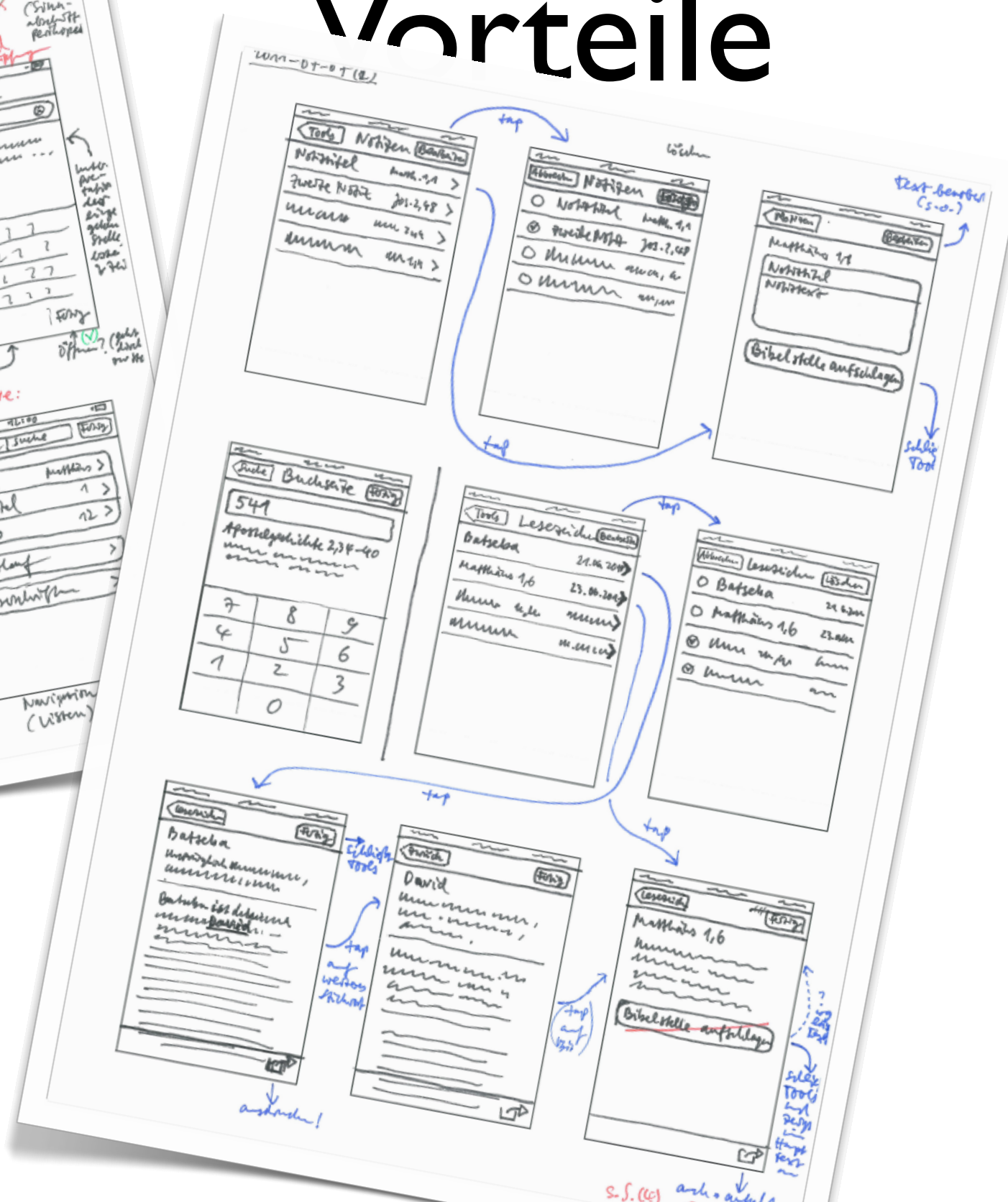
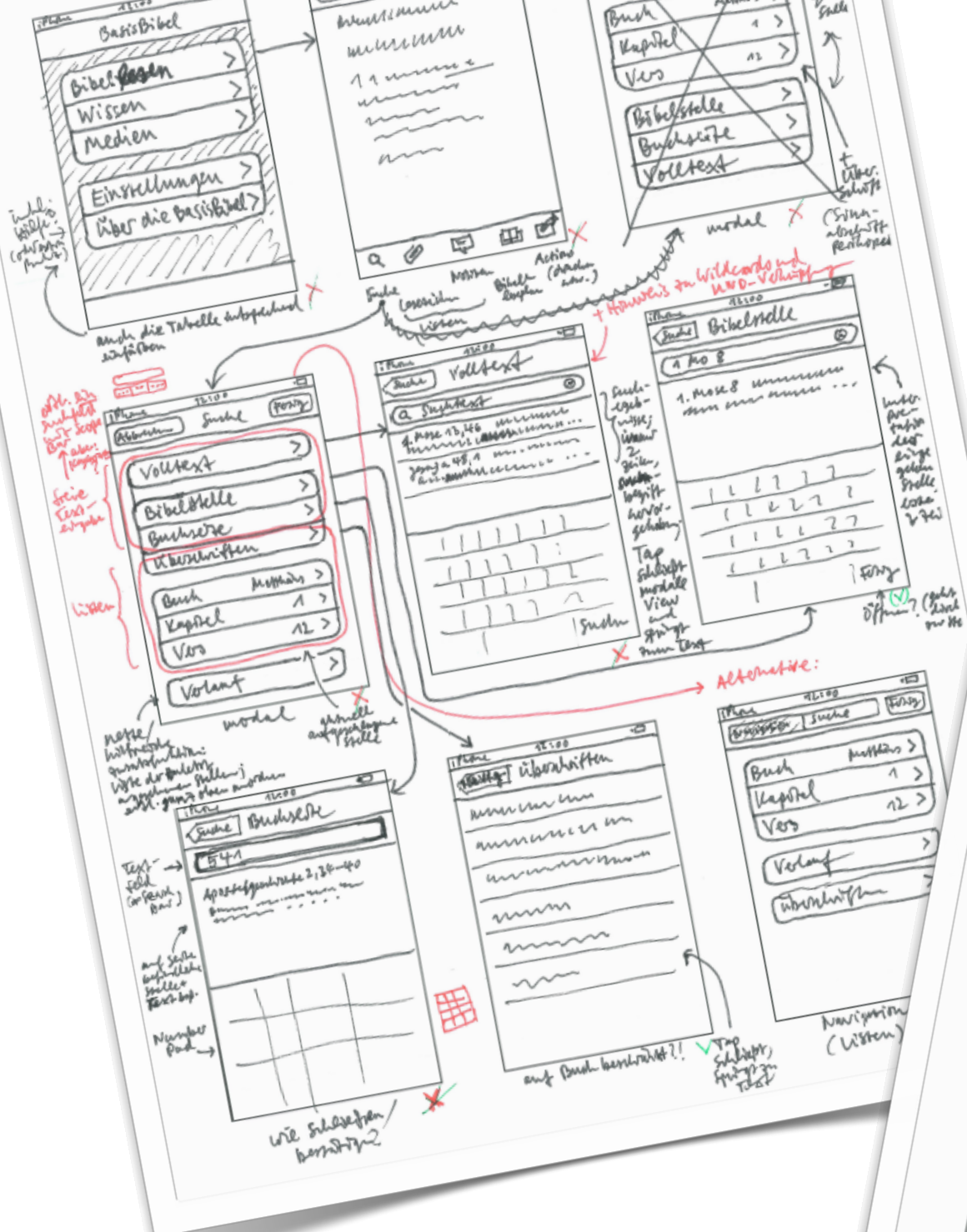
- Views in Code (z. B. OpenGL): View löschen, `-loadView`

# Vorteile

- Übersicht
- Interaktionen und Kontrollfluß verstehen
- Weniger Code
- Verbesserungen bei Table Views
- Rapid Prototyping



# Vorteile





# Nachteile

- Zunächst: einige Nachteile durch iOS 6 behoben
- nur iOS ( $\geq 5$ )
- Universal: Redundanz
- Schnell komplex (daher: Dekomposition)
- Wenig Layouthilfen
- Szenenliste scrollt nicht zur Auswahl

# Nachteile

- Versionskontrolle?
  - Aber: Edge Cases 22, »more human-readable XML than XIBs«

# Tips und Muster

- Dekomposition
  - Ein Storyboard pro Tab
  - Versionskontrolle
  - Schrittweises Konvertieren
- <http://robsprogramknowledge.blogspot.de/2012/06/linking-storyboard.html>
- <https://github.com/rob-brown/RBStoryboardLink>

# Tips und Muster

- Modal View Controller: wie kommt man zurück?
  - UINavigationController+Dismissal mit einer IBAction und  
`[self dismissViewControllerAnimated:YES completion:NULL]`
- <http://useyourloaf.com/blog/2012/10/08/presenting-view-controllers.html>
- Unwind Segues: recht kompliziert (Custom Segue usw.)

# Tips und Muster

- Einbetten verschiebt
- Kürzeste Pfade
- Freeform Size (Popover)
- Inferred/Explicit Metrics
- String-Konstanten für Identifier:  

```
extern NSString *const kMySegue           // .h  
NSString *const kMySegue = @"kMySegue";  // .m, IB
```

# Tips und Muster

- Plattformabhängiges Laden von Storyboards

```
UIUserInterfaceIdiom idiom = UI_USER_INTERFACE_IDIOM();
BOOL isPhone = (idiom == UIUserInterfaceIdiomPhone);
NSString *idiomSuffix = (isPhone) ? @"iPhone" : @"iPad";
NSString *storyboardName =
    [NSString stringWithFormat:@"Settings_%@", idiomSuffix];

UINavigationController *storyboard =
    [UINavigationController storyboardWithName:storyboardName
                                         bundle:nil];

id initialViewController =
    [storyboard instantiateInitialViewController];
```

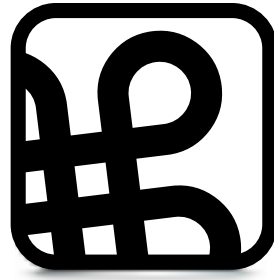
# Tips und Muster

- Gute Übersicht:  
<http://robsprogramknowledge.blogspot.de/2012/01/uistoryboard-best-practices.html>

Fragen?



**Vielen Dank**



**Macoun**