

Sascha Wolter | wolter.biz

Designer Developer Workflow

Prozesse, Technologien und Werkzeuge für Rich Applications und deren Auswirkung auf das Zusammenspiel von Gestaltern und Entwicklern.

Webinale 09, 26. Mai 2009

Über mich

- Sascha Wolter ist **Entwickler** und **Trainer** für Rich Applications mit dem Schwerpunkt **Flash/Flex Plattform** und **Silverlight**. Bereits seit **1995** arbeitet er als freiberuflicher **Autor**, **Berater** und **Software-Architekt**. Sascha ist ebenfalls durch zahlreiche internationale **Vorträge** auf führenden Konferenzen bekannt. Seine **Bücher** und **Artikel** rund um Flash, Flex und Silverlight zählen zu den meistgelesenen Publikationen. Als Gründer der führenden deutschsprachigen Adobe User Group flashforum.de mit rund **100.000 Mitgliedern** engagiert sich Sascha auch für die Belange der Anwender.
- **Letzte Veröffentlichungen**
 - [Video Training Flash CS3](#)
 - [Video Training Flex 3 Grundlage](#)
 - [Video Training Silverlight 2](#)
 - [Video Training ActionScript 3 Workshop](#)



Inhalt

- Zweieinhalb Welten
 - Verständnis und Kommunikation
 - Techniken und Werkzeuge (vom Wireframe zum Prototypen)
- Workflow
 - Nomenklatur
 - Trennung von Design und Code
 - Software-Architektur
 - Skinning und Templating
 - Enterprise
- Anhang
 - Fragen und Antworten

Designer und Developer Workflow

ZWEIEINHALB WELTEN

Zweieinhalb Welten: Verständnis!



„My Wife and My Mother-In-Law“ von William Ely Hill, 1915

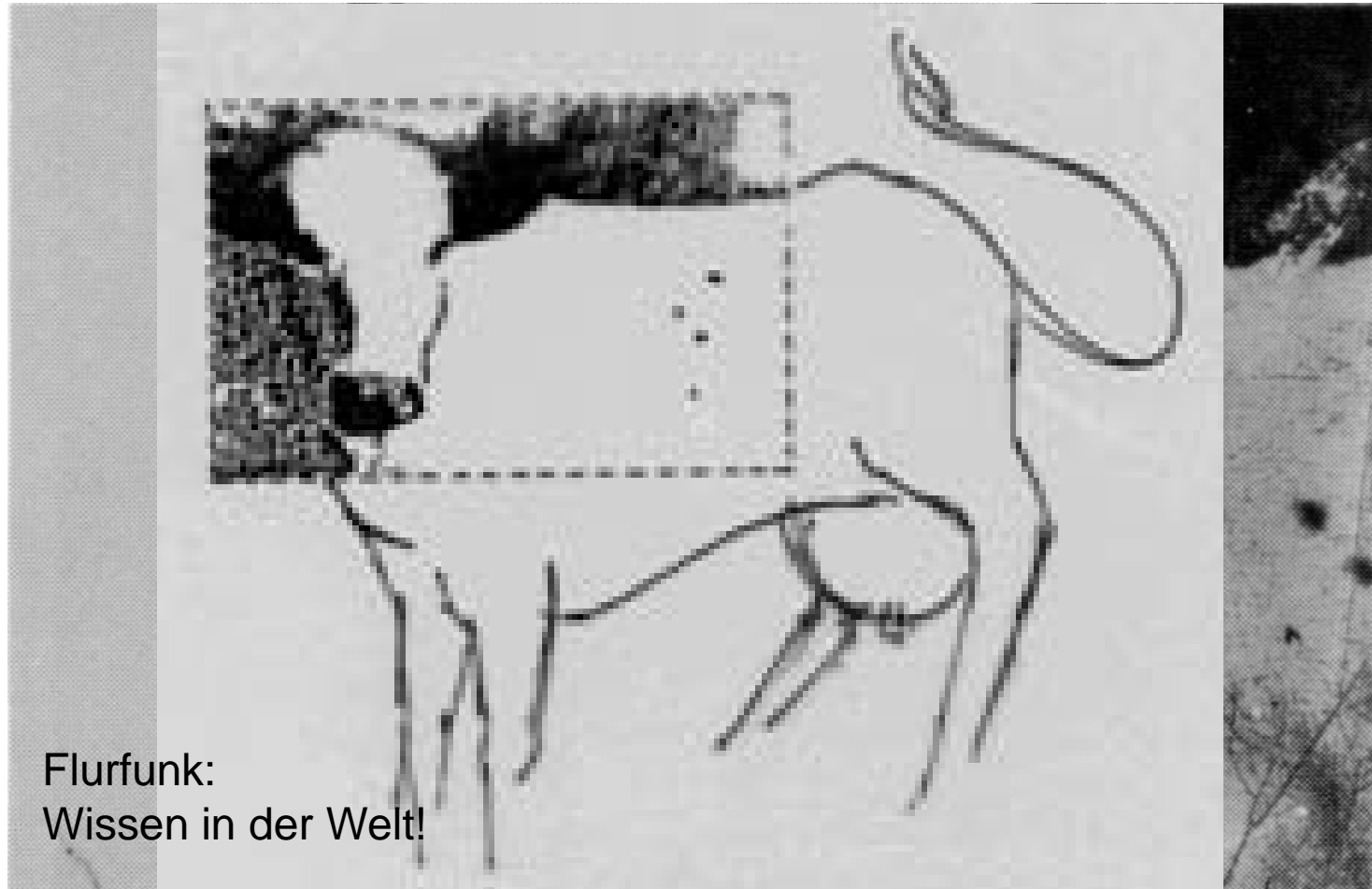
Verständnis

- Kontext
 - Vom gleichen sprechen!
 - Beispiel: 16:9 und Anamorph!
 - Workflow: Übergabe zwischen Designer und Developer
- Performance
 - Bandbreite und Darstellungsgeschwindigkeit
 - Vektoren versus Pixel
- Qualität
 - Skalierbarkeit: Die magische 4, 8 oder 16!
 - Beispiel: Was sind DPI!

Verständnis

- Team
 - Designer, Developer, Projektleiter, Kunde usw.
- Unterschiede
 - Ausbildung, Erfahrung, Sprache, Durchsetzungsvermögen usw.
- Herausforderung
 - Respekt
 - Kompetenz
- **Kommunikation**

Zweieinhalb Welten: Kommunikation!



Flurfunk:
Wissen in der Welt!

„Dallenberg Figure“ von Karl Dallenberg, 1952

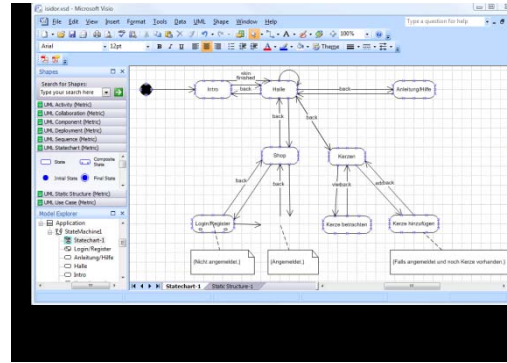
Kommunikation

- Klare Prozesse
- Klare **Aufgabenbereiche**
 - **Trennen von Design und Code:** Entwurfsmuster für Loose Coupling
- Klare **Schnittstellen**
 - **Zusammenbringen von Grafik und Implementierung:** Views und Skins
- Kreative Phasen versus Formale Phasen
 - Angelehnt an XP und agile Softwareentwicklung
- **Techniken versus Mediator**

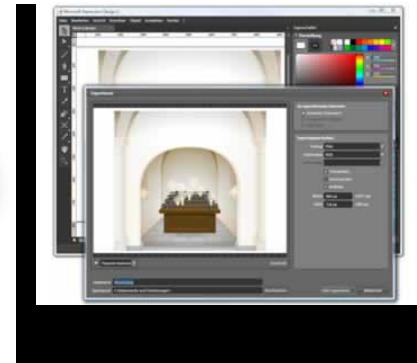
Phasen



- Konzept



- Planung
- Struktur

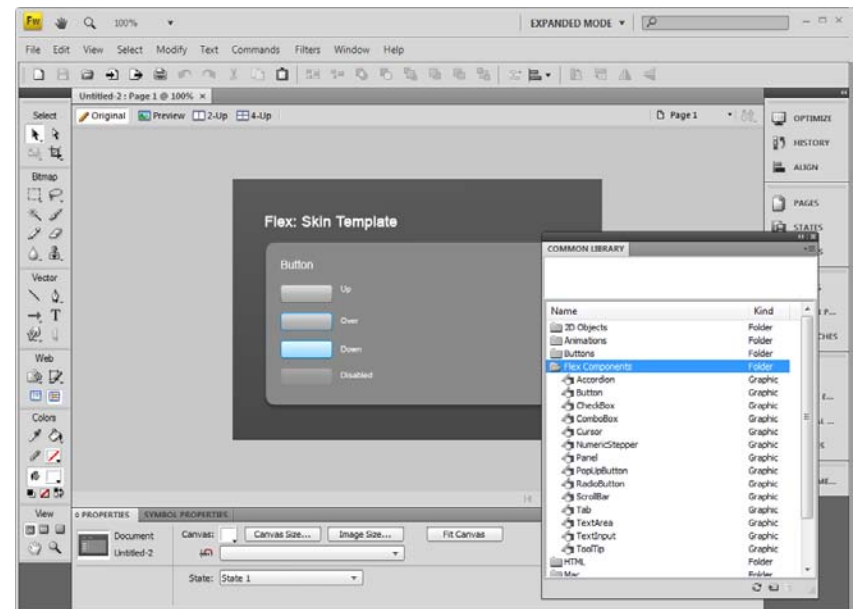


- Design
- Umsetzung



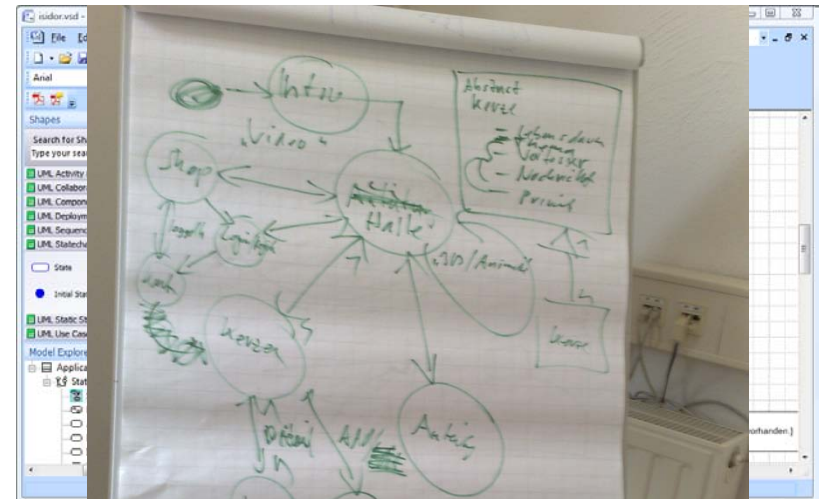
Techniken und Werkzeuge

- „Zustände der Anwendung (States)“
 - Für die Beschreibung einzelner Screens oder Views
- **Wireframe**
 - Fireworks: MXML, FXG (Flash XML graphics)* und „Skins“
 - ...
- **Komposition**
 - Photoshop: Skins
 - Illustrator: FXG und Skins
 - ...



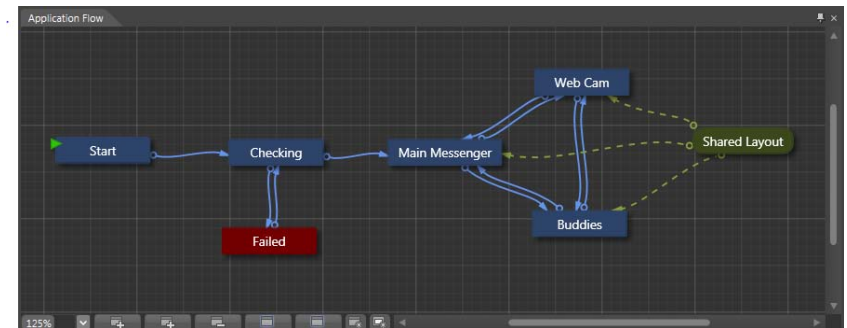
Techniken und Werkzeuge

- „Fluss der Anwendung“
 - Für die Beschreibung zusammenhängender Zustände und deren Abhängigkeit.
- Dope Sheet
 - Excel
- Storyboard
 - Powerpoint
- Sitemap
 - Dreamweaver
- **Zustandsdiagramm (UML)**
 - Visio
- Flussdiagramm
 - Visio



Techniken und Werkzeuge

- „Interaktiv“
 - Für die Beschreibung und den Test der Zustände und deren Interaktionen sowie deren Übergänge.
- Click-Dummy
 - Flash und Co.
- Mock-up
 - Karteikarten
- **Prototyping**
 - Axure RP pro (vom Wireframe zum Prototype)
 - Expression Blend (SketchFlow)



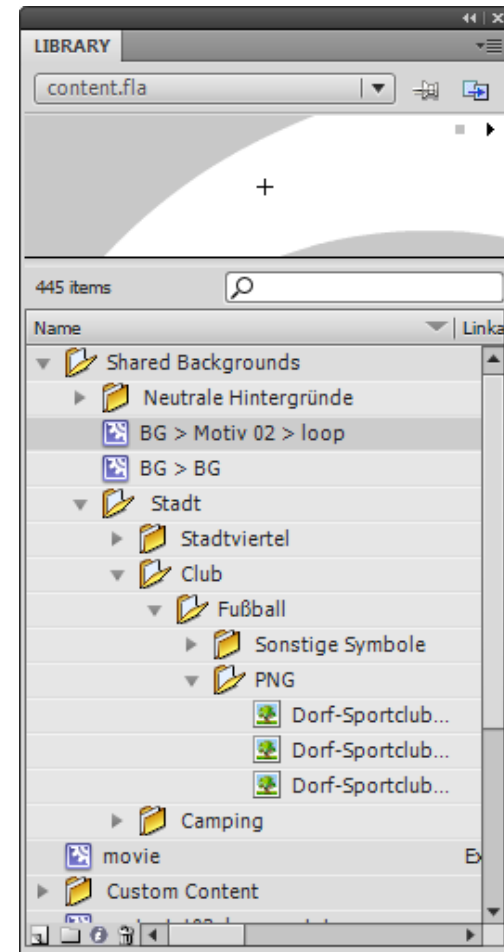
Designer und Developer Workflow

WORKFLOW

Nomenklatur

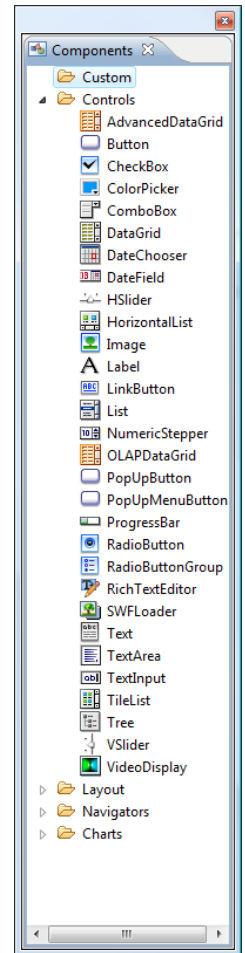
- **Bezeichnungen**

Bereits frühzeitig festlegen!
(z. B. im Wireframe)



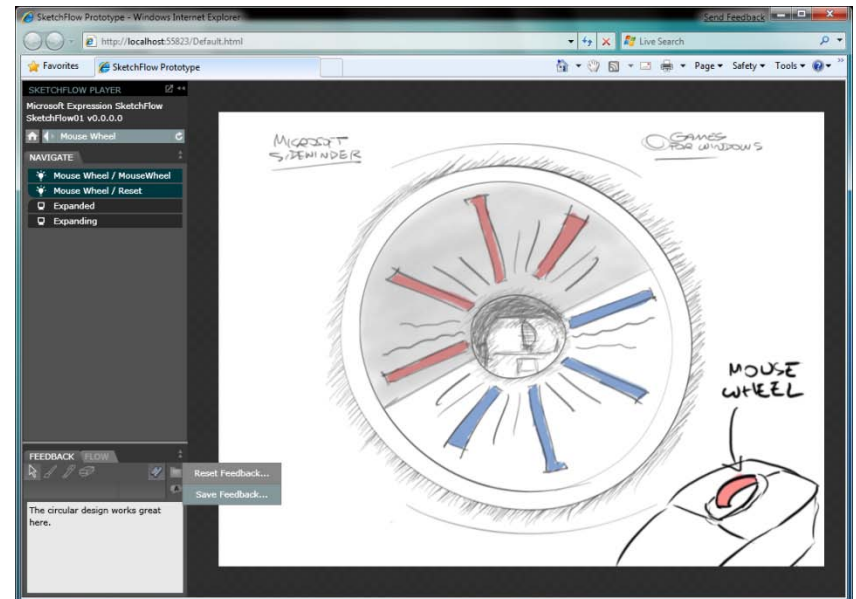
Nomenklatur

- **Baukasten definieren**
 - Wiederverwendbare Steuerelemente:
<http://silverlight.net/samples/sl3/toolkitcontrolsamples/run/default.html>
 - Gleichförmige Entwurfsmuster:
<http://quince.infragistics.com/>
- Stift und Papier und/oder Illustrator und Photoshop (**Komposition**)



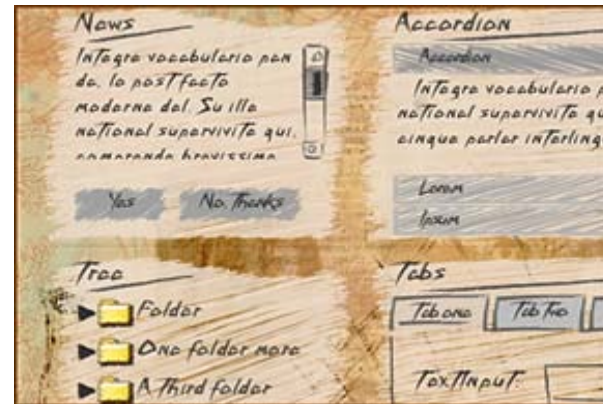
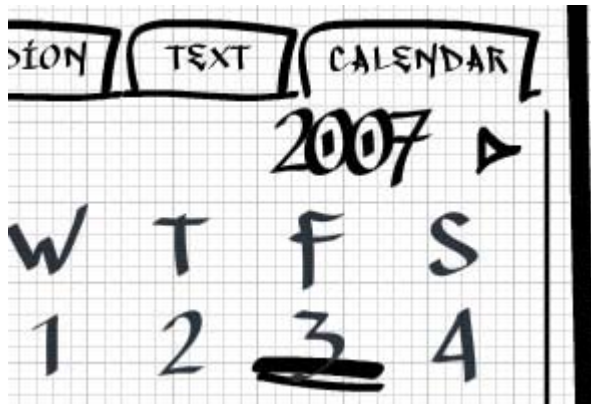
Nomenklatur

- Wireframe als Teil der Umsetzung
 - Mit Flash/Flex Builder
 - Flash Catalyst
 - Expression Blend



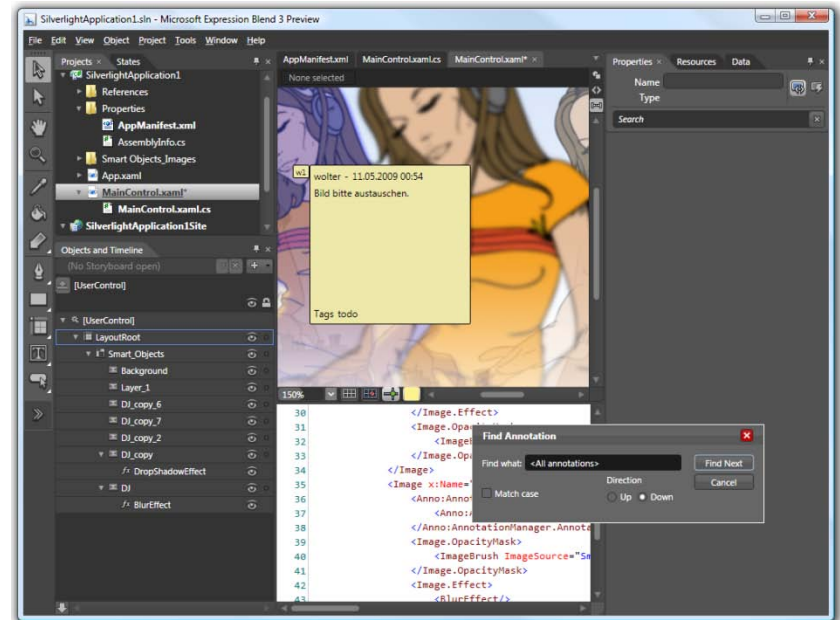
Nomenklatur

- Wireframe Theme
 - Damit es nicht **schon fertig** aussieht!
 - Runtime und Compiletime
 - Themes: http://fleksray.org/Flex_skin.html
 - `-theme=${flexlib}/themes/Wireframe/wireframe.swc*`



Trennung

- Design und Code trennen
 - Workflow und Wartbarkeit
 - Verhindern von verteiltem Code
- Hinweise im Design per Annotations

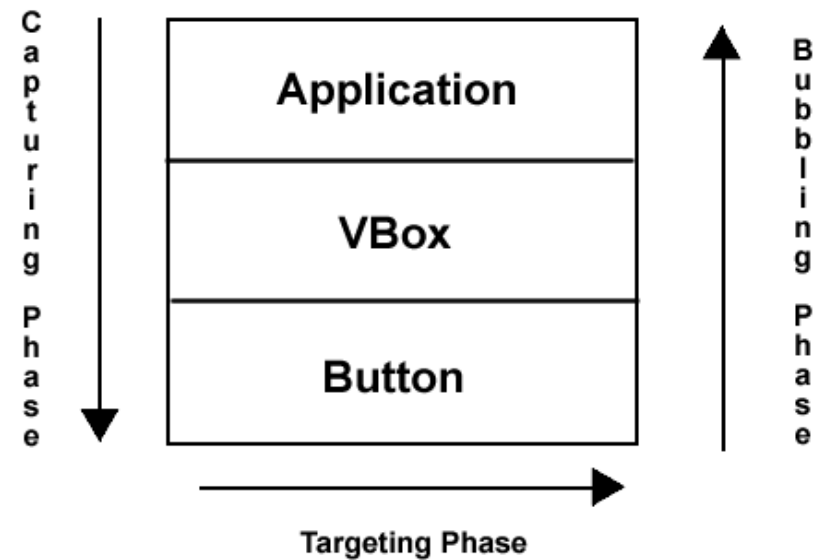


Trennung

- View versus Control
 - Ansichten ähneln Seiten
 - Ansichten sind konkret und mit einem konkreten Datenmodell assoziiert
 - Steuerelemente sind die Bausteine, aus denen die Seiten zusammengesetzt werden
 - Steuerelemente sind generisch und arbeiten mit Standarddatentypen
- Software-Architektur
 - Trennung des Views von der Geschäftslogik und den Daten
 - Loose Coupling

Trennung

- Entwurfsmuster
 - Code Behind
 - Model View Controller
 - Presentation Model
 - usw.
- Bindungen
 - oneway, twoway
- Ereignisse
 - Bubbling
 - Weiterreichen an das Modell



Skinning

Importiere Skin Assets nach Flex



Templates erleichtern die Erstellung von angepassten Komponenten-Skins



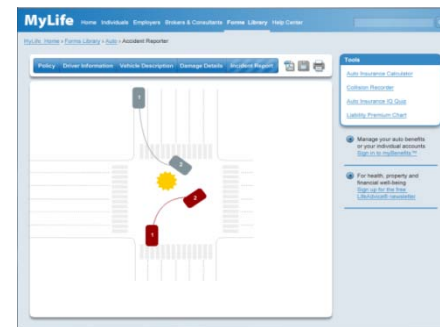
Import-Assistent erzeugt den Code, um die Assets den Komponenten zuzuweisen

MXML und CSS

Erzeuge Flex Komponenten in Flash



Flash-Developer können Inhalte einfach als Flex Komponente bereitstellen

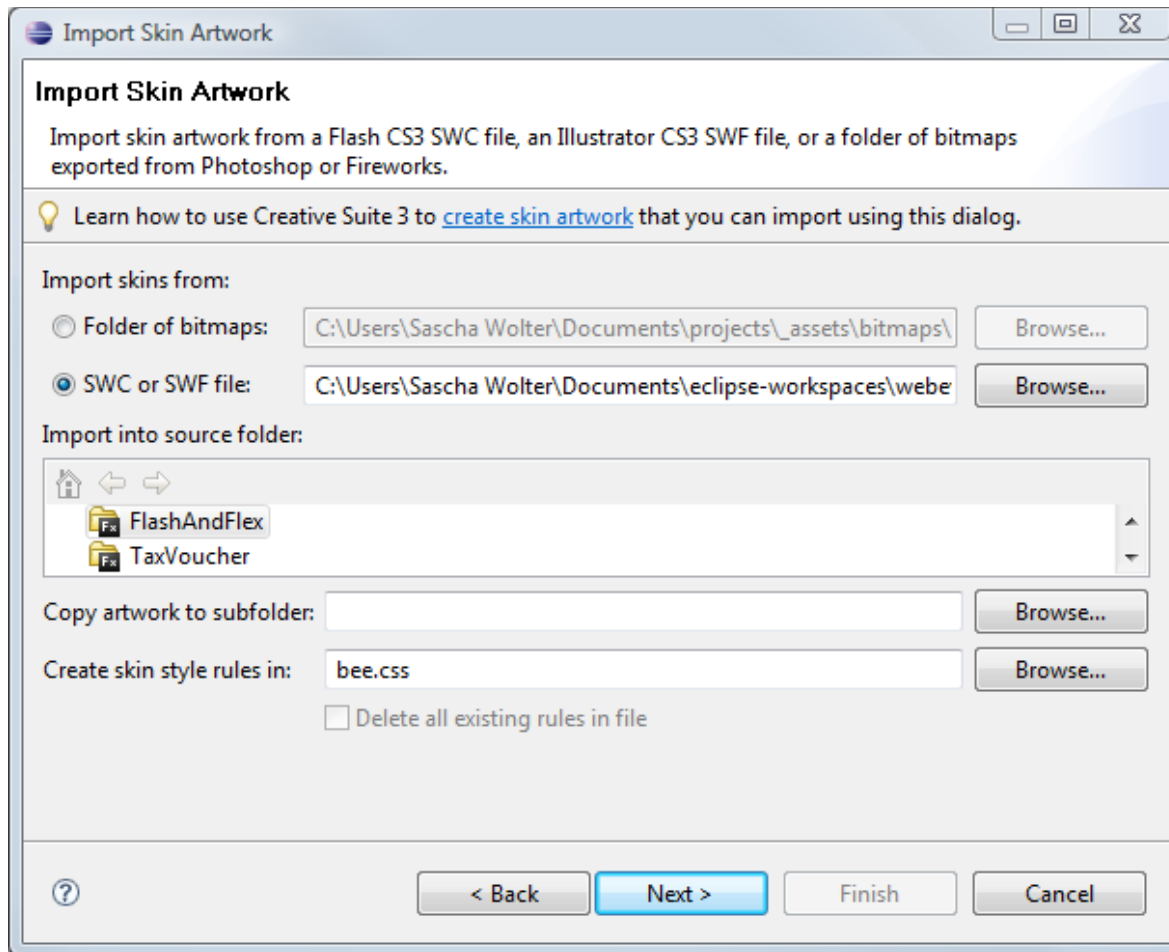


Skinning

- Styles versus Skins
 - Flex Style Explorer
 - Explorer: <http://designingwebinterfaces.com/great-flex-ux>
- Skinning
 - Kenntnis über die Komponente und deren Aufbau ist erforderlich
 - Vorgegebene Elemente können per Skin ausgetauscht werden
 - Angabe im MXML oder per CSS
 - Scale 9
- Theme
 - Sammlung von Styles und Skins
- Runtime und Compiletime CSS
 - Compile CSS to SWF, `StyleManager.loadStyleDeclarations("theme.swf");`
 - Library-Projekt mit CSS-Datei

Skinning

Skin Artwork



Skinning: Flash und Flex

- Flex Component Kit for Flash CS3/CS4
 - http://www.adobe.com/go/flex3_cs3_swfkit
 - Für die Nutzung von Flash in Flex
 - Inkl. Zeitleiste usw.
 - Sowohl in Flash als auch Flex nutzbar
 - Erzeugt eine Bibliothek (Library)
- Schritt für Schritt:
 1. Passende Version des Flex SDKs beachten;
 2. Erstelle Flash Inhalt als MovieClip Symbol mit erlaubtem Bezeichner und Registrierungspunkt oben links; Empfohlen sind 24 Bilder pro Sekunde;
 3. Prepariere Flash MovieClip Symbole für Flex über das Kommando “Make Flex Component” und veröffentliche anschließend ein SWC;
 4. Übertrage den Flash Inhalt nach Flex in dem die Bibliothek hinzugefügt wird;

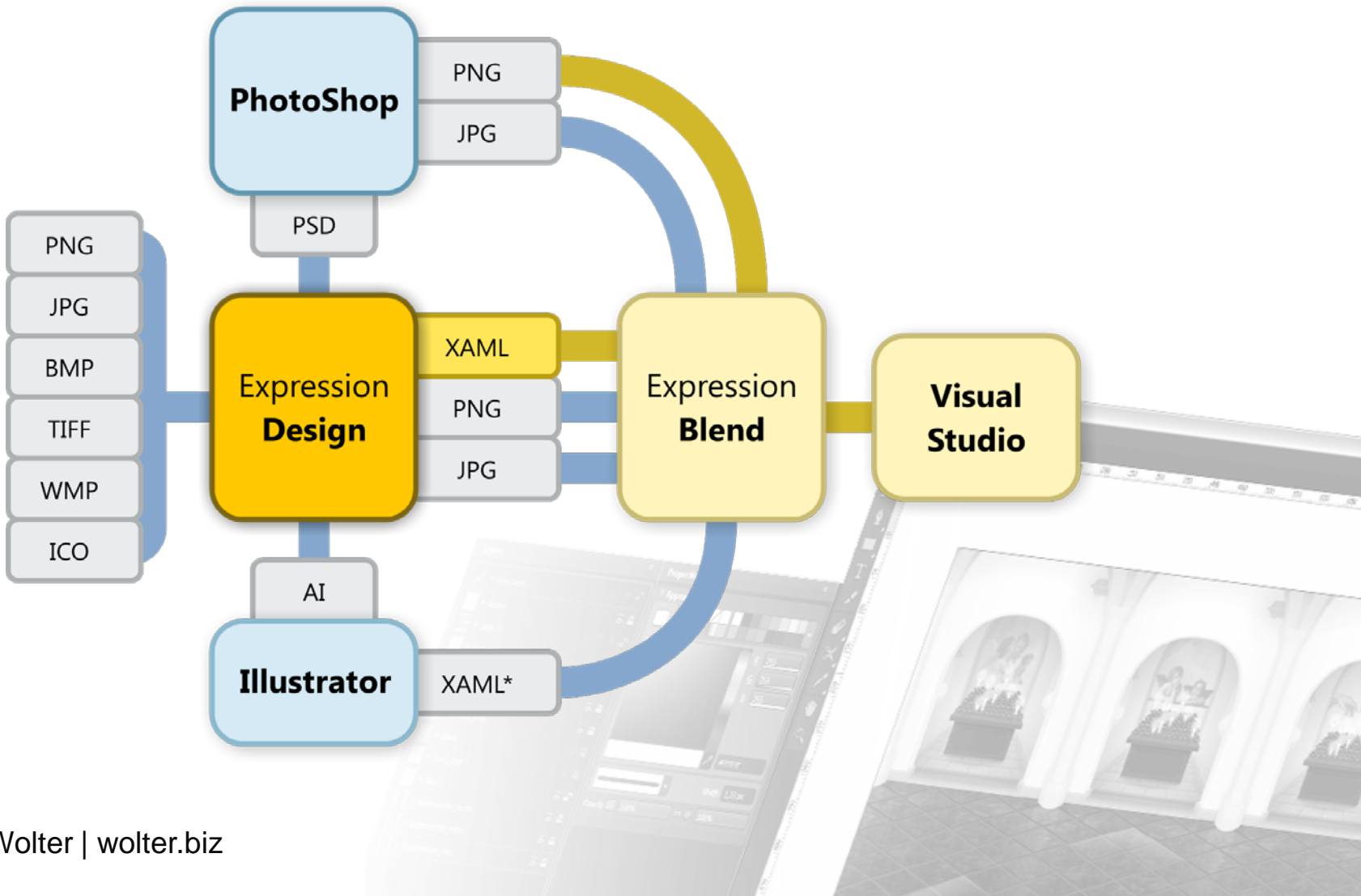
Skinning: Tipps

- Eigene Erweiterung (Flash)
 - MXP (Flash Extension)
 - z. B. JavaScript API
 - z. B. Komponenten
 - z. B. Templates
 - MXI
 - http://download.macromedia.com/pub/exchange/mxi_file_format.pdf
 - Extension Manager
 - http://www.adobe.com/de/exchange/em_download/
 - Packen
 - Installieren
- Degrafa: Declarative Graphics Framework
 - Für die deklarative Beschreibung eines Skins!
 - <http://www.degrafa.org/>

Templating

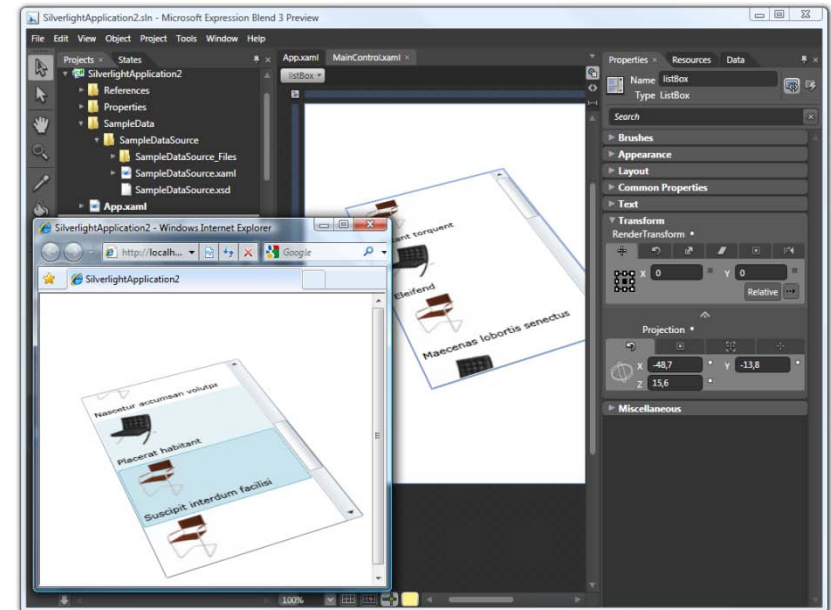
- **Vertrag** zwischen Design und Code
 - **States and Parts** (Zustände und Bestandteile)
 - Bindbare Eigenschaften
 - Gestaltung sehr frei, keine Kenntnis über die **Implementierung** notwendig
- Silverlight
 - wie WPF (Windows Presentation Foundation)
 - Styles und Templates
 - Tooling mit Expression Blend
- Flash/Flex
 - Erst mit Spark-Komponenten des Flex 4 SDK
 - Tooling mit Flash Catalyst

Templating



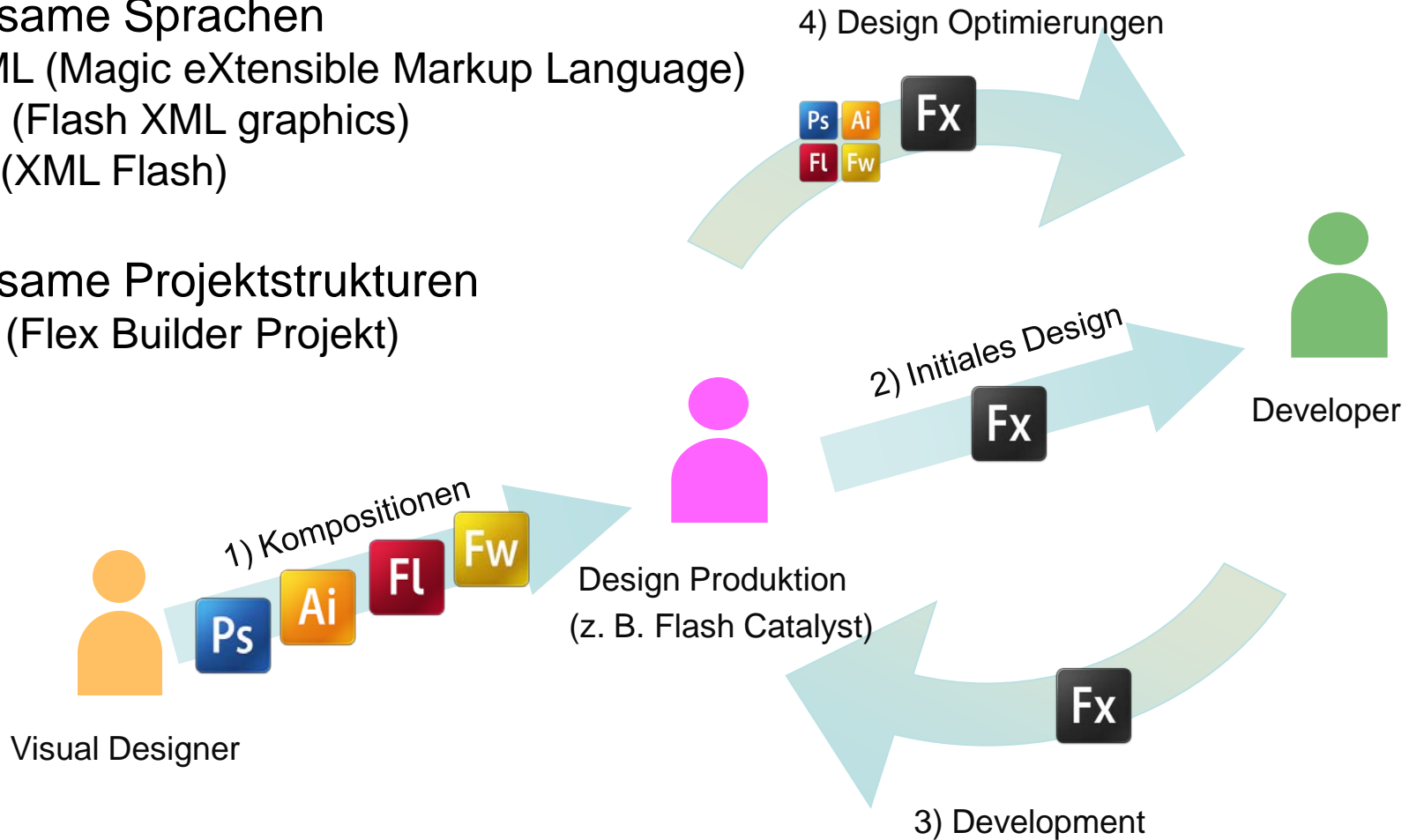
Zusammen: Enterprise-Umgebung

- Gemeinsame Umgebung
 - Gemeinsame Sprachen
 - Gemeinsame Projektstrukturen
 - Roundtrip
- Tests
 - Unit-Tests
 - Designzeitdaten
- Buildprozesse
 - Nightly Build
 - Continuous Integration



Zusammen: Flex/AIR

- Gemeinsame Sprachen
 - MXML (Magic eXtensible Markup Language)
 - FXG (Flash XML graphics)
 - XFL (XML Flash)
- Gemeinsame Projektstrukturen
 - FXP (Flex Builder Projekt)



Trennung: Mehrsprachigkeit

- Mehrsprachigkeit im Design
 - Beachten beim Layout
 - Nutzen zur Konfiguration
- Locale
 - Zur Kompilier- und zur Laufzeit
 - Property-Dateien, Resource-File
- ResourceBundle in Flex
 - `<mx:Label text="@Resource(key='helloWorld', bundle='Application')" />`
- Resource-File in Silverlight
 - `<Button x:Name="SaveBtn" Content="{Binding Path=saveBtnLabel, Source={StaticResource Resource}, Mode=OneTime}" />`
- Zahlreiche Datentypen möglich
 - `getClass`
 - Logo=Embed("logo.png")
 - Sorter=ClassReference("sorters.Urdu")

Designer und Developer Workflow

ZUSAMMENFASSUNG

Flash versus Silverlight



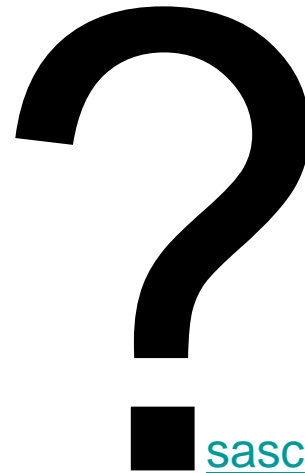
Konzepte
Werkzeuge
Interoperabilität

Ressourcen
Workflow
Anforderungen



Und ohne Mediator geht nichts?

Fragen und Antworten



sascha@wolter.biz

Literatur

- [Psychology of Everyday Things](#)
- [Don't Make Me Think](#)
- [Designing Web Usability](#)
- [Designing Interfaces](#)
- [Video Training Flex 3 Grundlage](#)
- [Video Training Silverlight 2](#)

Sascha Wolter | wolter.biz

Danke für Ihre Geduld

Designer Developer Workflow

Webinale 09, 26. Mai 2009