

Single Source Publishing mit Apache Forrest

by Ferdinand Soethe

Die Vorteile von XML basierter Dokumentation und Single Source Publishing sind in aller Munde, doch im Alltag dominieren noch immer Word, Framemaker und andere Apache Forrest ist ein Single Source Publishing Framework auf Basis von Apache Cocoon und XML-Standards.

Table of contents

1	Einleitung.....	3
1.1	Zu meiner Person	3
1.2	Themen heute.....	3
1.3	Unterlagen im Internet.....	3
2	Single Source Publishing kurz vorgestellt.....	3
2.1	Single Source Publishing?.....	3
2.2	SSP Komponenten.....	3
2.3	(Warum) Geht das nicht auch mit Word, Dreamweaver ...?.....	6
2.4	(Un)Sinn von WYSIWYG.....	6
3	Forrest kurz vorgestellt.....	7
3.1	Was will Forrest?.....	7
3.2	Forrest in Schlagworten.....	7
3.3	Forrest kennenlernen.....	7
3.4	Forrest-Strukturen.....	9
3.5	Forrest Features.....	12

3.6 Forrest als Framework.....	12
3.7 Think SSP!.....	13
3.8 SSP-Beispiel 1.....	14
3.9 Und die Programmierung?.....	14
3.10 Power SSP.....	14
4 Tools und Funktionalitäten.....	14
4.1 XML-Editoren.....	14
4.2 Revisions-Verwaltung.....	15
4.3 Forrest für Kleingruppen.....	15
4.4 Forrest Professionell.....	15
4.5 Staging mit ForrestBot.....	15
4.6 Freigabe Workflows.....	15
5 Forrest für Sie?.....	16
5.1 Forrest für Sie im Einsatz.....	16
5.2 Forrest Software Lizenz.....	16
5.3 Spielregeln für Open Source.....	16
5.4 Hilfe und Support.....	16
5.5 Wie zuverlässig ist Forrest eigentlich?.....	17

1 Einleitung

1.1 Zu meiner Person ...

Ferdinand Soethe

- Softwarearchitekt, Autor und Berater
- Mitglied des Apache Forrest PMC
- Arbeitsschwerpunkt: SSP-Systeme

1.2 Themen heute

- Das Konzept SSP erläutern,
- Zeigen, wie einfach SSP sein kann,
- (zum Beispiel mit Forrest),
- Sinnvolle SSP-Anwendungen,
- Tools und Methoden vorstellen.

1.3 Unterlagen im Internet

Die Unterlagen zu diesem Vortrag finden Sie im Internet unter

<http://soethe.net/vortraege/forrest>

Sie haben dort auch die Möglichkeit, Ihrer Meinung zu diesem Vortrag Ausdruck zu verleihen.

2 Single Source Publishing kurz vorgestellt

2.1 Single Source Publishing?

Kernidee des Single Source Publishing ist es, aus einem universellen Quelldokument verschiedene Medien zu bedienen.

Gut umgesetzt bedeutet das

- Gleichmässig hohe Qualität aller erzeugten Medien
- Autoren, die sich auf ihre eigentliche Arbeit konzentrieren können
- Kostenersparnis bei regelmäßigen Publikationen

2.2 SSP Komponenten

2.2.1 SSP Komponenten im Überblick

Single Source Publishing-Systeme bestehen in der Regel aus vier wichtigen Komponenten:

- Medienunabhängige Formatierung
- Auszeichnungsgrammatik
- (Geeignetes) Editierwerkzeug
- Publishing Framework

2.2.2 Medienunabhängige Formatierung

Medienunabhängige Formatierung (besser "*Auszeichnung*") ist die Grundlage jedes SSP-Systems.

Im Idealfalle zeichnet der Autor seine Texte nur dem Sinngehalt nach aus und hat / braucht keine Vorstellung davon, wie es später

- gedruckt aussieht,
- im Web dargestellt wird,
- sich anhört,
- riecht oder schmeckt!

2.2.3 Semantische Auszeichnung

Sinnbeschreibende Auszeichnungen sind eine gute Grundlage für SSP-Systeme.

Der Autor kennzeichnet den Text lediglich *dem Sinn nach* - zum Beispiel als "Zitat" - und überlässt alles weitere den medienspezifischen Ausgabemodulen.

```
Oft ist in der Literatur von  
  <Zitat> Single Source  
  Publishing mit Apache Forrest  
  </Zitat> die Rede.
```

2.2.4 Auszeichnungsgrammatik

Eine strikte und klar definierte Grammatik auf Basis von Schema oder Relax-NG unterstützt Autor wie Programmierer der Ausgabemodule in ihrer Arbeit.

2.2.5 (Geeignetes) Editierwerkzeug

Autoren profitieren besonders von Grammatik-tauglichen Editoren, die

- bei der Eingabe Elemente und Attribute vorschlagen
- und die Einhaltung der Grammatik sicherstellen.

Ideal sind hier WYSIWYN-Systeme geeignet, die sich aber nur sehr langsam durchsetzen.

2.2.6 WYSIWYN = Visuelles Editing

- **What-You-See-Is-What-You-Need** verbessert die Lesbarkeit von Texten durch visuelle Auszeichnungen.
- Es wird *nicht* das Ergebnis in einem bestimmten Medium dargestellt.
- Semantische Auszeichnungen werden durch Schrift, Farbe oder Prefixe visualisiert, die Auszeichnungen selbst werden i.d.R. unterdrückt.
- Mittlerweile bieten auch verschiedene kostengünstige Editoren (XMLMind, Oxygen) einen solchen Modus.

2.2.7 Publishing Framework

2.2.7.1 Aufgaben eines SSP-Frameworks

Das Publishing Framework bildet den Kern des SSP-Systems.

Wichtige Aufgaben:

- Quelltext überprüfen
- Quelltext transformieren
- Rahmen erzeugen

2.2.7.2 Quelltext überprüfen

Vor der Verarbeitung ist es sinnvoll, den Quelltext des Autors auf verschiedene Qualitätsmerkmale hin zu überprüfen:

- Einhaltung der Grammatik
- Verweis-Integrität
- Prüfung verwendeter Zeichensätze

2.2.7.3 Quelltext transformieren

Die Transformation des Quelltextes für das Ausgabemedium ist der Kern des SSP-Systems.

Im Rahmen des Framework-Konzepts können beliebige Transformationsmodule für weitere Medien hinzugefügt werden:

- Web-Seiten
- PDF
- Mindmaps
- ...

2.2.7.4 Rahmen erzeugen

Für vielen Medien muss der transformierte Quelltext zusätzlich in einen funktionalen Rahmen eingebettet werden:

- Header, Footer und Menüsystem bei Webseiten
- Allgemeine Angaben für PDFs

Gute SSP-Frameworks erledigen das automatisch.

Oft kann die "Rahmenhandlung" durch Konfiguration angepasst werden.

2.3 (Warum) Geht das nicht auch mit Word, Dreamweaver ...?

Konventionelle Publikationssysteme tun sich schwer mit echtem SSP,

- weil die Dokumentauszeichnungen häufig sehr medienspezifisch sind,
- weil sie Autoren zum visuellen Layout verführen,
- weil es sehr aufwändig ist, in einem Word-Dokument eine feste Grammatik durchzusetzen,
- weil zu viele Freiheiten in der Gestaltung die Verarbeitung für andere Medien schwer bis unmöglich machen.

2.4 (Un)Sinn von WYSIWYG

Zu bedenken ist auch,

- ob teuer bezahlte Fachleute wirklich ihre Zeit auf Textlayout verwenden sollten,
 - ob die Talente für visuelle Gestaltung und Schreiben nicht oftmals bei verschiedenen Menschen zu finden sind,
- (ganz zu schweigen von technisch hochwertigem Webdesign).

Ob also der Schritt zum wysiwyg als Vereinigung von Schreiben und Setzen nicht wirklich ein **Irrweg** war ?!

2.4.1 Additional Information

Eine ausführliche Diskussion über WYSIWYG und seiner Stärken und Schwächen findet sich unter [What has WYSIWYG done to us? \(http://www.ideography.co.uk/library/seibold/WYSIWYG.html\)](http://www.ideography.co.uk/library/seibold/WYSIWYG.html) von 1996. Darin schreibt der Autor unter anderem:

how we were seduced by WYSIWYG's illusion of control and how we lowered our expectations and typographic standards and became deeply confused about who in publishing is supposed to do what.

Und weiter:

Good typography requires a lot more than good-quality typefaces. It also requires improved composition algorithms within publishing software -- both for paper and for the Web.

Dieses Papier ist heute, gut zehn Jahre später noch immer sehr aktuell.

3 Forrest kurz vorgestellt

3.1 Was will Forrest?

Im Mission Statement auf der Forrest Homepage heisst es:

Apache Forrest is a publishing framework that transforms input from various sources into a unified presentation in one or more output formats.

The modular and extensible plugin architecture is based on Apache Cocoon and relevant standards, which separates presentation from content.

3.2 Forrest in Schlagworten

- SSP Publishing Framework
- Standard-basiert
- Unterstützung multipler Quellformate
- Trennung von Inhalt und Darstellung

3.3 Forrest kennenlernen

3.3.1 Forrest herunterladen

Die jeweils aktuelle Version von Forrest können Sie sich kostenlos von der Forrest Website (<http://forrest.apache.org/mirrors.cgi#closest>) herunterladen.

Zusätzlich brauchen Sie ggf. noch eine aktuelle Java Runtime in der Version 1.4 oder darüber.

3.3.2 Forrest installieren

Die Installation von Forrest ist ziemlich einfach und unter http://forrest.apache.org/docs_0_80/your-project.html#installing ausführlich dokumentiert:

1. Inhalte des ZIP-Archivs in ein beliebiges Verzeichnis entpacken
2. Zwei Umgebungsvariablen richtig setzen.

Fertig! Damit ist Forrest installiert und einsatzbereit.

3.3.3 Forrest Projekte anlegen

1. Befehlszeilenfenster öffnen
2. In ein neues, leeres Unterverzeichnis des Projektes wechseln
3. Forrest seed aufrufen

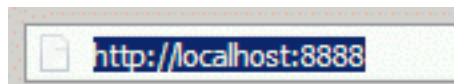
Forrest legt nun das Grundgerüst eines neuen Projektes für Sie an.

3.3.4 Dynamischer Modus

Das neue Projekt ist unmittelbar verwendbar.

Im dynamischen Modus (forrest run) läuft Forrest als Webserver (Application-Server) auf dem eigenen PC.

Wie einen Webserver sprechen wir ihn mit unserem Browser und einem bestimmten URL an ...



... und bekommen als Antwort eine dynamisch, d.h. in diesem Moment generierte Seite, unseres Projekts.

3.3.5 Leitseite eines neuen Projekts

Die Leitseite des neuen Projekts zeigt bereits alle Merkmale eines normalen Forrests:

myGroup > myProject >

MyGroup MyProje

Home Samples Apache XML Projects Plugins

Search the site with google Search

Last Published: 02/14/2007 09:28:53

▼ About

- ▣ Index
- ▶ Samples
- ▶ Plugins

built with Apache Forrest

Welcome to MyProj

PDF

- ▣ [Congratulatory](#)
- ▣ [Using examples as templates](#)

Congratulations

You have successfully generated and rendered an [Apache Forrest](#) site. This page is from the site template. It is found in

3.4 Forrest-Strukturen

3.4.1 Ein Blick auf Strukturen

Um Forrest zu verstehen, lohnt sich ein Blick in die Strukturen des neuen Projekts:

- Quelle der Seiten
- Navigationsstruktur
- site.xml
- Skins schaffen den Rahmen
- Individuelle Konfiguration
- CSS Formatierung

3.4.2 Quelle der Seiten

Im Normalfalle erzeugt Forrest Inhalte aus Quelldateien im Verzeichnisweig `src\documentation\content\xdocs`.

Hier ein Ausschnitt aus dem Quelltext der Leitseite `index.xml`

```
<!DOCTYPE document PUBLIC "-//APACHE//DTD Documentation V2.0//EN" "http://forrest.apache.org/dtd/document-v20.dtd">
<document>
  <header>
    <title>Willkommen beim Forrest Demoprojekt</title>
  </header>
  <body>
    <section
      id="status">
      <title>Congratulations</title>
      <p> You have successfully generated and rendered an <a
        href="ext:forrest">Apache Forrest</a> site. This page is
        from the site template. It is found in
        <code>src/documentation/content/xdocs/index.xml</code> Please
        edit it and replace this text with content of your own. </p>
    </section>
```

3.4.3 Navigationsstruktur

Grundsätzlich wird die Navigationsstruktur eines Forrest-Projekts durch zwei Dateien im `xdocs`-Verzeichnis definiert:

- `site.xml` enthält einen Eintrag für jede Seite, die als Menüpunkt sichtbar sein soll
- `tabs.xml` ordnet Seiten den Registern zu.

3.4.4 site.xml

Beispiel Leitseite

```
<index label="Index" href="index.html"
      description="Welcome to MyProj" />
```

Zusätzlich können Seiten zu Themenbereichen (Menüs) zusammengefasst werden:

```
<about label="About">
  <index
        label="Index" href="index.html" description="Welcome to
        MyProj" />
</about>
```

3.4.5 tabs.xml

tabs.xml definiert die Register ("tabs") im Kopf der Seite:

```
18 <!DOCTYPE tabs PUBLIC "-//APACHE//DTD Cocoon Documentation Tab V1.1//EN" "http://forrest.apache.org/dtd/tab-cocoon-v1.1.dtd">
19 <tabs software="MyProj"
20   title="MyProj"
21   copyright="Foo"
22   xmlns:xlink="http://www.w3.org/1999/xlink">
23   <tab id="" label="Home" dir="" indexfile="index.html"/>
24   <tab id="samples" label="Samples" dir="samples" indexfile="sample.html">
25     <tab id="samples-index" label="Index" dir="samples" indexfile="index.html"/>
26     <tab id="samples-sample2" label="Sample2" dir="samples" indexfile="static.html"/>
27   </tab>
28   <tab label="Apache XML Projects" href="http://xml.apache.org">
29     <tab label="Forrest" href="http://forrest.apache.org"/>
30     <tab label="Xerces" href="http://xml.apache.org/xerces"/>
31   </tab>
32   <tab id="plugins" label="Plugins" dir="pluginDocs/plugins_0_80" indexfile="index.html"/>
33 </tabs>
```

3.4.6 Skins schaffen den Rahmen

Skins (engl. "Häute") schaffen den Rahmen, in den die Seiten eingebettet werden.

Bei Bedarf kann leicht ein anderes Skin installiert und verwendet werden; hier zum Beispiel unsere Leitseite mit einem ganz einfachen Skin:

Congratulations

You have successfully generated and rendered an [Apache Forrest](#) site. This page is from the site template. It is found in `src/documentation/content/xdocs/index.xml`. Please edit it and replace this text with content of your own.

Using examples as templates

This demo site has many examples. See the menu at the left. The sources for these examples are in the directory `src/documentation/content/xdocs/`

The sources for the Apache Forrest website are also included in your distribution at `$FORREST_HOME/site-author/`

3.4.7 Skin wechseln

Sie können jederzeit leicht ein anderes Skin installieren:

1. Den Server vorübergehend beenden.
2. Der Befehl `forrest available-skins` erzeugt eine Liste verfügbarer Skins.
3. In der Datei `forrest.properties` die Zeile

```
#project.skin=pelt
```

> suchen und die neue Zeile

```
project.skin=[NeuerName]
```

darunter einfügen.

4. Mit `forrest install-skin` das neue Skin aktivieren.
5. Den Server erneut starten.

Übrigens: Bei Bedarf können Sie Forrest auch parallel mit verschiedenen Skins betreiben!

3.4.8 Individuelle Konfiguration

Manche Forrest-Skins sind von vornherein konfigurierbar gestaltet, so dass Sie viele Elemente ohne Programmierung verändern können.

Die meisten Optionen finden sich in der Datei `src\documentation\skinconf.xml`:

- Soll die Seite parallel als PDF angeboten werden?
- (Wie) sollen mail-Adressen verschleiert werden?
- Welche Logos sollen im Kopf der Seite erscheinen?
- Für welche Papiergröße sollen die PDFs erzeugt werden?
- Was steht in den Fußzeilen der PDFs?
- (Wo) wird ein lokales Menü aller Kapitel einer Seite angezeigt?

(Aus Platzgründen nennen wir hier nur einige wenige Beispiele.)

3.4.9 CSS Formatierung

Da das aktuelle Skin "pelt" eine saubere CSS-Formatierung verwendet, bietet sich hier eine weitere mächtige Eingriffsmöglichkeit:

Mit nützlichen Werkzeugen wie der Web Developer Extension für Firefox (<http://chrispederick.com/work/webdeveloper/>), können CSS-Änderungen am lebenden System ausprobiert und sodann im Abschnitt <extra-css> der skinconf.xml dauerhaft ins Projekt integriert werden.

3.5 Forrest Features

Schon in der Standard-Anwendung liefert Forrest ein nützliches SSP-System mit vielen Vorteilen:

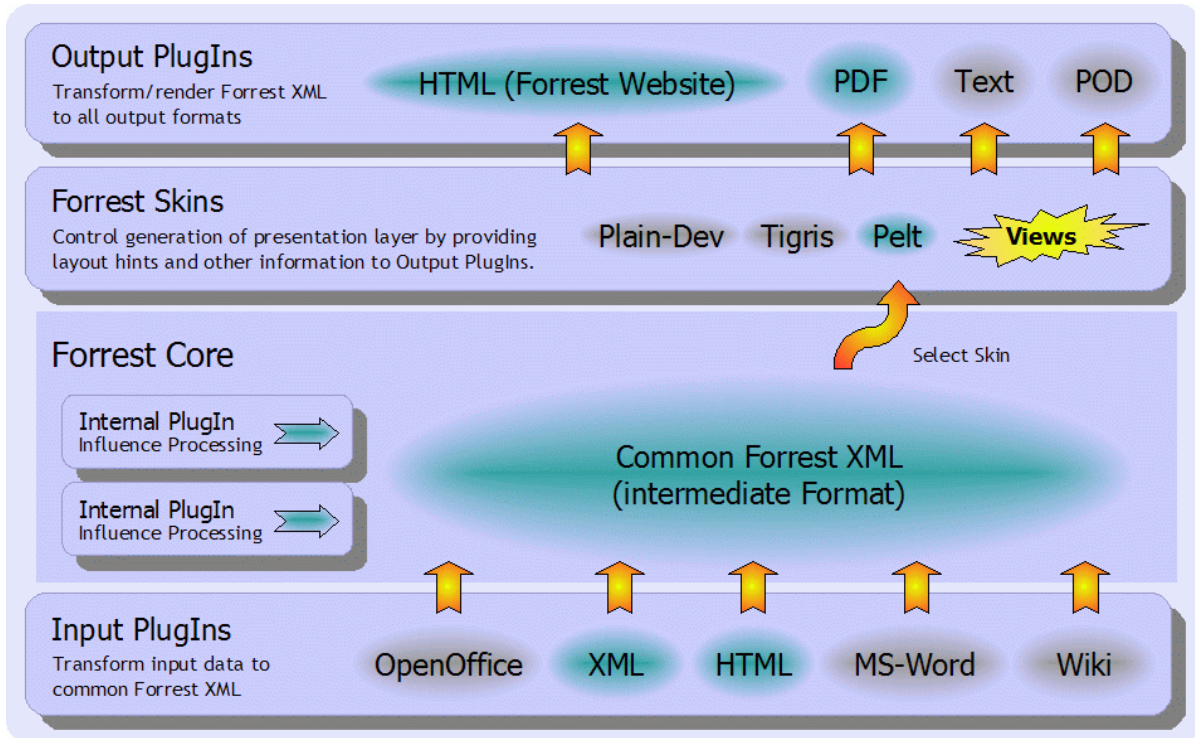
- Wahlweise dynamische oder statische Generierung von Inhalten
- Automatische Erzeugung hochwertiger PDFs
- Barrierearmes HTML
- Navigationssystem, das mit und ohne JavaScript funktioniert
- Saubere Trennung von Inhalt und Präsentation

3.6 Forrest als Framework

3.6.1 Framework = mehr als eine Anwendung

Forrest ist jedoch als Framework, also erweiterbar, konzipiert worden und wächst daher bei Bedarf mit.

3.6.2 Offene PlugIn-Architektur



3.6.3 Input PlugIns

Mit Input-PlugIns können zahlreiche weitere Quellformate in ein Forrest-System integriert werden.

Zur Zeit unter anderem Html, XHTML, Simplified DocBook, Wiki und OpenDocument.

Auch dynamische Daten, wie RSS-Feeds, können als Quelle verwendet werden.

3.6.4 Output PlugIns

Output-PlugIns erweitern die Ausgabe-Optionen um weitere Medien wie PDF, POD, Text oder s5-Präsentationen.

3.6.5 Interne PlugIns

Auch die Verarbeitung selbst kann mit PlugIns beeinflusst werden, zum Beispiel um entsprechend markierte Worte in ein Glossar aufzunehmen.

3.7 Think SSP!

3.8 SSP-Beispiel 1

Die Startseite eines Websites soll - neben statischem Text - regelmässig auch 10 aktuelle News-Texte enthalten.

Das kann man als normale HTML-Seite gestalten (und damit die Gestaltungsarbeit auf die Redakteurin abwälzen).

Mit SSP und XML geht das aber besser:

- Eine spezielle Grammatik "erfindet" die Homepage als feste Struktur.
- Eine Transformation wandelt diese in Forrest in HTML um, wobei die Programmierung dafür sorgt,
 - dass nur die zehn aktuellsten Punkte verwendet werden,
 - dass sie absteigend chronologisch sortiert sind und
 - dass immer genau die CSS-Formatierung verwendet wird, die für die News geplant ist.

3.9 Und die Programmierung?

Wer keine Ahnung von XML, Schema oder XSL-T mitbringt, muss anfangs viel investieren. Oder jemanden beauftragen.

- Das genannte Beispiel: 4 Stunden
- RSS-Feed für News: 1 Stunde

3.10 Power SSP

SSP eröffnet eine neue Welt im Umgang mit Dokumenten:

- Mit Aufgaben-spezifischen Grammatiken kann die Erstellung von Dokumenten stark beschleunigt und zugleich qualitativ gesichert werden.
- Verarbeitung in Transformationen erlaubt multi-dimensionale Filterung
- Ergebnisse können vielseitig verwendet werden.
- Aus Dokumenten können sogar interaktive Anwendungen werden.

4 Tools und Funktionalitäten

4.1 XML-Editoren

Der XML-Editor ist der kritische Punkt für jedes System.

- Das Eclipse-Webtools-Paket bietet eine sehr brauchbare Open Source-Lösung mit grammatikfähigem Editor und Revisionsverwaltung als Plugin.

- oXygen <http://www.oxygenxml.com/> ist eine hochwertige Lösung für Programmierung und Authoring (aber kommerziell!).
- XMLMind <http://www.xmlmind.com/xmleditor/>
- Amaya (www.w3.org/Amaya/) und BitFlux (bitfluxeditor.org/) sind auf dem richtigen Weg aber noch zu instabil.

4.2 Revisions-Verwaltung

Forrest selbst enthält keine Revisionsverwaltung.

Sie kann aber mit den Open Source-Werkzeugen *Subversion* (<http://subversion.tigris.org/>) und *Tortoise* (<http://tortoisesvn.tigris.org/>) leicht nachgerüstet werden.

4.3 Forrest für Kleingruppen

Forrest ist ein dateibasiertes Werkzeug und nur begrenzt Multiuser-fähig.

In kleineren Gruppen und mit einem Editor, der bearbeitete Dateien sauber sperrt (Locking), kann ein Forrest-Projekt auch mit mehreren Personen bearbeitet werden.

Empfehlenswert ist der Einsatz von Subversion mit optimistischem Locking-Mechanismus und sehr guter Konfliktverarbeitung.

4.4 Forrest Professionell

Für professionelles Arbeiten in Gruppen empfiehlt sich eine Server-basierte Installation:

- Forrest als Anwendung in einem echten Application-Server wie *Apache Tomcat* (<http://tomcat.apache.org/>)
- *Subversion*-Server zur zentralen Verwaltung des Repositories
- *Tortoise* als lokaler Subversion-Client unter Windows

4.5 Staging mit ForrestBot

ForrestBot (<http://forrest.apache.org/tools/forrestbot.html>) realisiert automatisches oder halbautomatisches "Staging", also die Veröffentlichung eines Forrest-Projekts als statische Seiten auf einen ftp- oder anderen Server.

4.6 Freigabe Workflows

Forrest ist selbst kein CMS und unterstützt das Arbeiten mit Freigabe-Workflows nicht.

Allerdings kann man bei Einsatz von Subversion und Forrestbot eine ähnliche Funktionalität erreichen:

- Autoren bearbeiten Artikel und checken sie bei Abschluss mit Subversion ein.

- Die Redakteurin wird (auf Wunsch) von Subversion informiert und kann Änderungen sichten, korrigieren oder auch zurücksetzen.
- Zu einem beliebigen Zeitpunkt können in Subversion Releases erstellt und veröffentlicht werden.

5 Forrest für Sie?

5.1 Forrest für Sie im Einsatz

Im Folgenden noch paar Tipps für den Start mit Forrest:

5.2 Forrest Software Lizenz

Forrest steht unter der Apache Software Lizenz.

Das bedeutet:

- Dass Sie Forrest ohne Lizenzgebühren nutzen dürfen
- ... auch für kommerzielle Zwecke (und mit sehr wenigen Einschränkungen).
- Selbst wenn Sie auf Basis von Forrest etwas neues entwickeln, muss das nicht zwingend Open Source Software werden.

Allerdings: Sie erhalten Forrest 'ohne Garantie und Gewährleistung'.

5.3 Spielregeln für Open Source

Open Source Software-Entwicklung basiert auf freiwilliger Mitarbeit!

Daher gelten hier andere Regeln als bei "normaler" Software:

- Sie können kostenfrei nutzen, was es schon gibt,
- und es gerne weiter verbessern.
- Sie können auch jemanden engagieren, der es für Sie erledigt!

5.4 Hilfe und Support

Wie bei den meisten Open Source Projekten gibt es keine Forrest Support Hotline mit hauptamtlichen Mitarbeitern und keinen Anspruch auf Support ...

- ... doch auf der Mailing Liste (<http://forrest.apache.org/mail-lists.html>) erhalten Sie fast immer schnell und unbürokratisch Antworten auf Ihre Fragen.
- Darüber hinaus sind bei Bedarf auch Support-Verträge in deutscher Sprache erhältlich.
- Und wenn ein Fehler zum Problem wird ...
 - helfen Sie uns, ihn zu beheben!
 - Sie werden sehen, dass andere versuchen werden, Ihnen dabei zu helfen.

5.5 Wie zuverlässig ist Forrest eigentlich?

Meiner Erfahrung nach ist Forrest eine verlässliche Plattform.

Allerdings ist es sinnvoll, ein paar Grundregeln zu beachten.

- Stellen Sie einen Teil der eingesparten Lizenzgebühren als Budget für Verbesserungen beiseite.
- Verwenden Sie einen weiteren Teil auf vernünftige Ausbildung Ihrer Mitarbeiter
- Nehmen Sie sich Zeit um Forrest in Ruhe einzuführen!
- Helfen Sie uns, Forrest weiterzuentwickeln.