

Best Practices for Plone in Higher Ed

Ploneability 2007

Author: Joel Burton <joel@joelburton.com>

Copyright: Copyright 2007 Joel Burton. ALL RIGHTS RESERVED.

Contents

1	Best Practices for Plone in Higher Education	6
1.1	About Me	7
1.1.1	About Me	7
1.1.2	My Goals	7
1.1.3	Where Are You?	7
1.2	Choosing Plone	8
1.2.1	Choosing Plone	8
1.2.2	Getting the Support	8
1.3	Building a Site	9
1.4	Talents Required	10
1.4.1	HTML	10
1.4.2	What Requires HTML	10
1.4.3	What Doesn't Require HTML	10
1.4.4	CSS	11
1.4.5	Python	11
1.4.6	What Requires Python?	11
1.4.7	What Doesn't Require Python?	12
1.4.8	UML	12
1.4.9	Geek Pythoneer	12
1.5	Training	13
1.5.1	Training Expectations	13
1.5.2	Pre-Training	13
1.5.3	Core Training	13
1.5.4	Scripting	14
1.5.5	Developing	14
1.5.6	Deployment Strategies	14
1.5.7	Consulting	15
1.5.8	Ongoing Expectations	15
1.5.9	"Continuing Ed" Training	15
1.6	Add-On Products	16

1.6.1	Add-On Products	16
1.6.2	How to Choose	16
1.6.3	How to Choose (2)	17
1.6.4	Recommended	17
1.7	The Short List	18
1.7.1	The Short List	18
1.7.2	University Products	18
1.8	Strategies for Success	19
1.8.1	Single Site or Per Department?	19
1.8.2	Linking Separate Sites	19
1.8.3	Kupu	19
1.8.4	Versioning / Staging	20
1.8.5	Single Sign-On / Shared PW	20
1.8.6	Layered Theming	20
1.8.7	IRC	21
1.8.8	User Groups	21
1.8.9	Reaching Me	21
2	Building a Site	22
2.1	Structuring Site	23
2.1.1	How to Structure Site	23
2.1.2	How to Structure Site (2)	23
2.1.3	How To Structure Site (3)	24
2.2	Strategy	25
2.2.1	Building a Plone Site	25
2.2.2	What Types of Content?	25
2.2.3	What Kind of Security/Workflow?	26
2.2.4	How Should Content Look?	26
2.2.5	What Other Features?	26
2.2.6	How Should Site Look?	27
2.2.7	Deployment Considerations	27

3	Web Services	28
3.1	Introduction	29
3.2	Web Service	30
3.3	XMLRPC	31
3.3.1	XMLRPC	31
3.3.2	Example Web Service	31
3.3.3	XMLRPC in Perl	31
3.3.4	XMLRPC in Java	32
3.3.5	Sample Script	32
3.3.6	Testing	32
3.4	SOAP	33
3.4.1	SOAP Overview	33
3.4.2	SOAP Client	33
4	Add-On Products	34
4.1	Overview	35
4.1.1	Overview	35
4.1.2	How to Choose	35
4.1.3	How to Choose (2)	36
4.1.4	Recommended	36
4.2	The Short List	37
4.3	Membership	38
4.3.1	Users / Authentication	38
4.3.2	Users / Authentication (2)	38
4.3.3	Membership	38
4.3.4	Membership (2)	39
4.4	Content Types	40
4.4.1	Discussion Boards	40
4.4.2	Form Generation	40
4.4.3	Blogs	40
4.4.4	Documentation	40
4.4.5	Project Management	41

4.4.6	General Content Types	41
4.5	Archetypes Support	42
4.5.1	Fields and Widgets	42
4.5.2	Fields and Widgets (2)	42
4.5.3	Fields and Widgets (3)	43
4.5.4	Date/Time	43
4.5.5	Schemas	43
4.6	Internationalization	44
4.7	Services	45
4.7.1	Reporting	45
4.7.2	Version Control	45
4.7.3	Services	46
4.7.4	Deployment	46
4.7.5	Management	46
4.7.6	Commerce	47
4.8	Catalog	48
4.9	Development Tools	49
4.9.1	Development	49
4.9.2	Debugging	49
4.9.3	Debugging (2)	49
5	Footnotes	50

1 Best Practices for Plone in Higher Education

1.1 About Me

1.1.1 About Me

- Plone trainer, integrator
 - Developer of “Plone Bootcamp”
 - Taught to over 45 universities
 - * Secret keeper of the plots
-

1.1.2 My Goals

- How to choose Plone
 - Building site process
 - Talent & Training
 - Stumbling Blocks & Strategies
-

1.1.3 Where Are You?

- Are you ...
- ... evaluating Plone?
- ... building your first site?
- ... supporting your first site?
- ... supporting multiple sites?

1.2 Choosing Plone

1.2.1 Choosing Plone

- Can't circumvent the process
 - Pick us for the *right* reasons
 - Standards compliance
 - Separation of concerns
 - Secure by default & stays that way
-

1.2.2 Getting the Support

- We have "good names"
- **Security:** CIA
- **Scalability:** Novell
- **Geek cred:** Discover Magazine, NIH
- **Domain:** Penn State University, Harvard

1.3 Building a Site

Building a Plone Site

- What types of content?
- What kind of security/workflow?
- How should content look?
- What “other features” are needed?
- How should site look?
- Deployment considerations

1.4 Talents Required

1.4.1 HTML

- Hopefully, widely-available :)
 - Comfort with hand-code required
 - * But WYSIWYG tools will work great
-

1.4.2 What Requires HTML

- New templates
 - New content types
 - Overall site design
-

1.4.3 What Doesn't Require HTML

- Routine content creation / editing
 - Some is always useful for cleanup
- Form development

1.4.4 CSS

- Often widely available
 - But often not very sophisticated:
`.blue { color: blue; }`
 - We get 100% of our look from CSS
 - And you'll want to, too
-

1.4.5 Python

- Difficult to find already
 - Surprisingly easy to pick up
 - Especially as 2nd “scripting language”
 - Excellent for occasional use
-

1.4.6 What Requires Python?

- “Application” development
 - Specialized processes

1.4.7 What Doesn't Require Python?

- Form development
 - PloneFormGen can solve this
 - Specialized content types
 - Can build with ArchGenXML
 - or `uml.joelburton.com`
 - Workflow development
 - Can build in UML or TTW
-

1.4.8 UML

- Used for custom content type creation
 - And possibly, workflow creation
 - Not widely known in web-design world
 - More common in Comp Sci
 - A tiny bit will help a lot!
-

1.4.9 Geek Pythoneer

- Often, only need one or two
 - Widespread reasonable scripting more in demand

1.5 Training

1.5.1 Training Expectations

- Plone is a **big** product
 - Three comprehensive stacks
 - Multiple “ideologies”
 - Good but not perfect books
 - Realistically, training is required
-

1.5.2 Pre-Training

- 3-4 days of self-play
 - Get comfortable with Kupu, concepts, etc.
 - Recommended for everyone
 - Can be self-directed with some goals
-

1.5.3 Core Training

- How the parts work
- “Shared technologies”
 - ZPT
 - Lightweight Python (expressions, etc)
- 4-5 days for scripters/templaters/programmers

1.5.4 Scripting

- Building small apps / new content types
 - Focus on Python + moving parts
 - Often can be done in “mentoring mode”
 - 4-5 days (assuming they have Python experience)
-

1.5.5 Developing

- Building significant apps or extending core functionality
 - Python-focused
 - 15-20 days
 - Few training programs cover this
 - Tends to be customized/mentoring/self-learning
-

1.5.6 Deployment Strategies

- Caching (several parts)
- Testing
- Good for scripters + developers
- 1-2 days
 - Often better done later

1.5.7 Consulting

- Security
 - At least for an audit!
 - Server setup/config
-

1.5.8 Ongoing Expectations

- “New product analysis”
 - Who’s watching the collective?
 - * Upcoming review site may help
 - 1d/month
-

1.5.9 “Continuing Ed” Training

- Often happens at conferences
 - Or advanced bootcamps ;)
- Expect at least 1w/year

1.6 Add-On Products

1.6.1 Add-On Products

- Almost all products are on plone.org
 - Over 450!
 - Some excellent, some dead-ends
 - Most migration problems come from 3rd party products
-

1.6.2 How to Choose

- Generally, ignore product name
 - Anything can be called “PloneThisOrThat”
- Generally, ignore version numbers
 - Since they’re inconsistent
- Generally, ignore “developer buzz”
 - Since their agendas/skills are different

1.6.3 How to Choose (2)

- Look at the author
 - Some have a reputation for excellence/support
 - Ask around
 - On IRC or at user group

And consider the skill level of the person making the recommendation. A recommendation from an advanced developer may not be as useful as one from an experienced integrator.
 - See if they support old/new versions
 - A product that doesn't reach backwards may not reach forwards
-

1.6.4 Recommended

- Useful products for Plone
 - Generally, “best of breed”
 - All should work with Plone 2.5

1.7 The Short List

1.7.1 The Short List

- DocFinderTab
 - Clouseau
 - CMFEditions (part of Plone 3)
 - RedirectionTool
 - PloneFormGen
 - LinguaPlone
-

1.7.2 University Products

- TeamSpace: team collaboration
- ScriptableFields: blurring the separation
- Enfold Desktop: nontech content management
- PloneRSSSearch: cross-site searching
- ManagedIndex and AdvancedQuery
- LinguaPlone (no, really!)
- Webservices (built-in)

1.8 Strategies for Success

1.8.1 Single Site or Per Department?

- Most universities are choosing per-department
 - For scalability/flexibility reasons
 - * Often politically related to rollout
-

1.8.2 Linking Separate Sites

- Cross-site searching
 - Using PloneRSSSearch or similiar strategies
 - Can “stuff” catalog with brains for external
 - Can delegate to 3rd party search
-

1.8.3 Kupu

- A pain point for many users
 - Not a kupu problem, but a control problem
 - * Win the battle on the merits
 - * Or fall back to fckEditor or others
 - But still win the war!

1.8.4 Versioning / Staging

- Handled out-of-the-box in Plone 3
 - CMFEditions for versioning
 - iterate for staging
 - Both can work with Plone 2.5
-

1.8.5 Single Sign-On / Shared PW

- Lots of “PAS Plugins” available
 - Including CAS, LDAP, etc
 - Easy to write new ones
 - Assuming you know underlying tech
 - Can Apache do this for you?
 - We can trust Apache
-

1.8.6 Layered Theming

- University has look which varies per department
 - Often handled with “layered theme products”
 - WebLion group has growing product for TTW themeing

1.8.7 IRC

- Hands down, best support option
 - Also, get connected in community
 - Consider your own channel
 - Keeps best practices internal
 - Builds community
-

1.8.8 User Groups

- Excellent way to connect to students
 - And local community
 - Universities make the best hosts
 - Consistent space, access to pizza
 - Easier than it looks & never seen one fail
-

1.8.9 Reaching Me

- Feedback, strategies welcome!
 - joel@joelburton.com
- Handouts from this talk
 - <http://plonebootcamps.com/resources>

2 Building a Site

2.1 Structuring Site

2.1.1 How to Structure Site

- Member's folder:

`http://site.org/Members/joel/project-a`

- "My Documents" feeling
 - Bad URLs
 - * What happens when ownership moves?
 - Wrong location or broken URL
 - Useful for community sites
-

2.1.2 How to Structure Site (2)

- Folder of like content:

`http://site.org/events/party`

- Requires users to have permissions here
- Useful when one person/department manages items of that type

2.1.3 How To Structure Site (3)

- By project / authority:

- `http://college.edu/chemistry`

- `http://college.edu/biology`

- Location should follow security

- Generally, the most useful

- Type-specific folders within:

- `http://college.edu/chemistry/events/party`

2.2 Strategy

2.2.1 Building a Plone Site

- What types of content?
 - What kind of security/workflow?
 - How should content look?
 - What “other features” are needed?
 - How should site look?
 - Deployment considerations
-

2.2.2 What Types of Content?

- Standard Plone content types
 - News items, events, HTML pages, etc.
- Downloadable content types from plone.org
 - Blogs, forums, magazine articles, etc.
- Make new content types with UML/Archetypes
- Clone content types for new “meaning”
 - News Item -> Press Release

2.2.3 What Kind of Security/Workflow?

- Standard Plone publishing
 - Friendly, cooperative
 - Your custom business workflow
 - Different workflows for different content types
-

2.2.4 How Should Content Look?

- What's "inside the content well"?
 - Write PageTemplates for your content types
-

2.2.5 What Other Features?

- Special applications
 - Shopping carts, maps, special searching
- Investigate products at plone.org
- Write your own with scripts or as "product"
 - Can be a very small or very big project

2.2.6 How Should Site Look?

- Optimally, HTML is for structure
 - So your site doesn't "look like" much
 - You can turn off styles to see "naked Plone"
 - CSS is used to make the HTML pretty
 - Not Plone-specific, but requires CSS skills
-

2.2.7 Deployment Considerations

- Where will users be stored? (Plone? LDAP?)
- Where will Plone store data? (ZODB? others?)
- Do you want a staging server?
- Make caching/proxying decisions
- Create functional tests with Selenium

3 **Web Services**

3.1 Introduction

Overview

- What is a web service?
- XMLRPC
- SOAP

3.2 Web Service

What is a Web Service?

- Way for a program to ask question of web site
 - Get simple answer, not full HTML page
 - Answer can be a number, string, list, etc.
 - * Isn't always a string, like normal HTTP

3.3 XMLRPC

3.3.1 XMLRPC

- “XML Remote Procedure Calls”
 - Straightforward protocol for web services
 - Similar to SOAP, but a bit simpler
-

3.3.2 Example Web Service

- Using standard Python XMLRPC library:

```
import xmlrpclib
url = "http://plonebootcamps.com/courses/sf"
service = xmlrpclib.Server(url)
print service.getCourseCost()
```

- Returns cost of class as real integer
 - Same in other languages (Java, C, etc.)
-

3.3.3 XMLRPC in Perl

```
#!/usr/bin/perl

use Frontier::Client;

$url = "http://plonebootcamps.com/courses/sf";
$client = Frontier::Client->new(url=>$url);
print $client->call('getCourseCost');
```

3.3.4 XMLRPC in Java

```
import org.apache.xmlrpc.XmlRpcClient;
import java.util.Vector;

public class CourseCost {
    public static void main(String args[])
        throws Exception {
        XmlRpcClient client = new XmlRpcClient(
            "http://plonebootcamps.com/courses/sf");
        Object result = client.execute("getCourseCost");
        System.out.println(result);
    }
}
```

3.3.5 Sample Script

- postalcodes.py:

```
## Parameters=postalcode

places = { '94114': 'San Francisco',
           '20036': 'Washington DC' }

return places.get(postalcode, '')
```

- Can be used as web service without changes
-

3.3.6 Testing

- Can test that in a regular python script:

```
import xmlrpclib
url = 'http://localhost:8080/Plone'
server = xmlrpclib.Server(url)
print server.postalcodes('94114')
```

3.4 SOAP

3.4.1 SOAP Overview

- A more comprehensive web service protocol
 - Not built-in to Zope/Plone
 - Easily added with SOAPSupport product
 - * then Zope calls handled by SOAP
-

3.4.2 SOAP Client

- SOAP client in Python:

```
from SOAPpy import SOAPProxy
url = 'http://localhost:8080/Plone'
server = SOAPProxy(url)
print server.postalcodes(url)
```

4 **Add-On Products**

4.1 Overview

4.1.1 Overview

- Almost all products are on plone.org
 - Over 450!
 - Some excellent, some dead-ends
 - Most migration problems come from 3rd party products
-

4.1.2 How to Choose

- Generally, ignore product name
 - Anything can be called “PloneThisOrThat”
- Generally, ignore version numbers
 - Since they’re inconsistent
- Generally, ignore “developer buzz”
 - Since their agendas/skills are different

4.1.3 How to Choose (2)

- Look at the author
 - Some have a reputation for excellence/support
 - Ask around
 - On IRC or at user group

And consider the skill level of the person making the recommendation. A recommendation from an advanced developer may not be as useful as one from an experienced integrator.
 - See if they support old/new versions
 - A product that doesn't reach backwards may not reach forwards
-

4.1.4 Recommended

- Useful products for Plone
 - Generally, "best of breed"
 - All should work with Plone 2.5

4.2 The Short List

The Short List

- DocFinderTab
- Clouseau
- CMFEditions
- RedirectionTool
- PloneFormGen
- LinguaPlone

4.3 Membership

4.3.1 Users / Authentication

- LDAPUserFolder
 - Members/memberdata in LDAP
 - SQLPASPlugin
 - Members/memberdata in rel dbs
 - NoDuplicateLogin
 - Prevent multiple logins
-

4.3.2 Users / Authentication (2)

- AutoRole
 - Give roles automatically on conditions
 - Inside firewall?
 - After 5pm?
 - Has a certain cookie set?
-

4.3.3 Membership

- Membrane / Remember
 - Members can act like content
 - * New fields
 - * Membership can be workflowed
 - Complex and not well documented yet

4.3.4 Membership (2)

- TeamSpace
 - Members can join “teams”
 - Teams admins manage team members
 - Team membership can be workflowed
 - Complex and not well documented yet

4.4 Content Types

4.4.1 Discussion Boards

- Ploneboard
-

4.4.2 Form Generation

- PloneFormGen
 - Top-10 product!
 - Content managers can create forms
 - * Email results
 - * Or add to database
-

4.4.3 Blogs

- Quills
 - Reasonably featureful
 - Recent versions have been getting more complex
 - * Be careful
-

4.4.4 Documentation

- PloneHelpCenter
 - Store FAQs, HOWTOs, etc
 - Used to power Plone’s “Documentation Center”

4.4.5 Project Management

- Poi
 - Bug-tracking software
 - PloneSoftwareCenter
 - Track software releases and projects
-

4.4.6 General Content Types

- RichDocument
 - “Article”-type of content
 - Stores images and files inside

I often use for magazine articles.
- RichTopic
 - Like SmartFolder, but with a rich body field

4.5 Archetypes Support

4.5.1 Fields and Widgets

- ArchAddOn
 - Learn to make new fields & widgets
 - A nice learning product.
 - ATExtensions
 - Useful fields for table-fields and more
 - Wicked
 - Lightweight, wiki-like linking syntax
 - Will ship with Plone 3.0
-

4.5.2 Fields and Widgets (2)

- ExternalStorage
 - Store content field in SQL DBs
- FCKEditor
 - Use FCKEditor instead of Kupu
- DateBox
 - Smarter, better date widget

4.5.3 Fields and Widgets (3)

- Scriptable Fields
 - Body fields can contain TAL/TALES
 - Fields that are Python scripts

Breaks some of the content/skin/script separation.

4.5.4 Date/Time

- CalendarX
 - “Microsoft Outlook” style views of calendar
 - * Monthly, daily, etc.
-

4.5.5 Schemas

- ATSchemaEditorNG
 - Content managers can customize Archetypes’ schemas
 - Useful for “local” customizations

4.6 Internationalization

Internationalization

- **LinguaPlone**
 - Maintain your content in different languages
 - Excellent!

4.7 Services

4.7.1 Reporting

- LinkChecker
 - Reports on broken outbound links
 - RedirectionTool
 - Allow moving of content without breaking links
 - ATRatings
 - Rate content
-

4.7.2 Version Control

- CMFEditions
 - Excellent product!
 - View old versions of content
 - * Can diff to current version
 - * Can rollback to old version

4.7.3 Services

- CMFSin
 - Show RSS feeds in portlets
- PloneWorkflows
 - Additional workflow policies

Included in Plone 3.0.

- PloneSEO
 - Control search engine settings for content
-

4.7.4 Deployment

- Entransit
 - Use Plone to manage content, other systems to deploy

Useful if you want to deploy with PHP, Java, etc.

Have to rebuild some dynamic features in the deployment language.

- ArcheCSV
 - Import content from CSV files
-

4.7.5 Management

- EnfoldDesktop
 - Manage Plone content via Windows desktop
 - Commercial product but inexpensive

4.7.6 Commerce

- Plone GetPaid
 - New framework for ecommerce
 - Still pre-alpha, but interesting

4.8 Catalog

New Catalog Indexes

- `TextIndexNG`
 - Very full-featured full-text search
- `FieldedTextIndex`
 - Search across indexes & weight indexes
- `ManagedIndex`
 - Advanced searching capabilities

4.9 Development Tools

4.9.1 Development

- CallProfiler
 - What scripts/templates are taking so long?
 - PTProfiler
 - Detailed info about ZPT performance
 - FSDump
 - Dump ZODB skin objects to disk
-

4.9.2 Debugging

- DocFinderTab
 - Ask any object how to work with it
 - Clouseau
 - Python prompt for Plone
 - An excellent product
-

4.9.3 Debugging (2)

- WingIDE
 - Commercial debugger/editor
 - Best debugger for Zope/Plone

5 Footnotes