

edi.substanceforms

Projektarbeit Seppo Walther

Seppo Walther

Creative Commons (CC BY-ND)

Inhaltsverzeichnis

1. Technische Dokumentation von edi.substanceforms	3
1.1 Zusammenfassung	3
1.2 Technische Voraussetzungen für den Einsatz von edi.substanceforms	3
1.3 Installation von edi.substanceforms für das Web-CMS Plone	3
2. Ausgangssituation	4
3. Zielsetzung	5
4. Anforderungsanalyse	6
4.1 Bedienbarkeit über das Web-CMS Plone	6
4.2 Responsive Webdesign	6
4.3 Barrierefreiheit gemäß BITV 2.0 (WCAG 2.1 Level AA)	7
4.4 Integrität der gespeicherten Daten	7
5. Bestandteile der Applikation (Module und Klassen)	8
5.1 Artikeltypen für die Pflege von Inhalten	8
5.2 Formulare	8
5.3 Views	8
5.4 Funktionale Views	9
6. Aufbau des Software-Packages	10
7. Automatische Testverfahren	11
7.1 Scope der Funktionstests	11
7.2 Test-Verzeichnis im Package	11
7.3 Ausführung der Tests	12
8. Installation und Konfiguration im CMS Plone	13
9. Lizenzvereinbarung für edi.substanceforms	14
10. Anhang	15
10.1 Test 1 2 3 4 5	15

1. Technische Dokumentation von edi.substanceforms

Die Software und die vollständige Online-Dokumentation finden Sie hier: edi.substanceforms.

Autor: Seppo Walther (seppo.walther@educorvi.de)

1.1 Zusammenfassung

Mit dem Add-On edi.substanceforms für das freie Web-Content-Management-System [Plone](#) wird die Online-Datenbank **Emissionsarme Produkte** der Berufsgenossenschaft Energie Textil Elektro und Medienerzeugnisse (BG ETEM) realisiert.

Mit Hilfe dieser Online-Datenbank können Unternehmerinnen und Unternehmer, Arbeitsschützer und Verantwortliche in den Betrieben der Branche Druck und Papierverarbeitung Chemikalienprodukte finden deren Einsatz in der Produktionsumgebung mit möglichst geringen Gesundheitsgefahren für die damit in Kontakt stehenden Beschäftigten verbunden ist.

Die Entwicklung des Plone-Add-On war Gegenstand der Projektarbeit von Seppo Walther im Rahmen der Ausbildung zum Fachinformatiker für Anwendungsentwicklung und wurde dem Prüfungsausschuss der IHK Nürnberg für Mittelfranken zur Bewertung vorgelegt.

Auch wenn es unwahrscheinlich erscheint, dass die hier entwickelte Software 1:1 in anderen Umgebungen oder für vergleichbare Anwendungsfälle eingesetzt werden kann, so liefert das Projekt doch verschiedene Hinweise und Softwarebestandteile für vergleichbare Projekte. Es wurde daher entschieden, das Projekt als Beispiellösung auf der Plattform [Github] zu veröffentlichen.

1.2 Technische Voraussetzungen für den Einsatz von edi.substanceforms

- Lauffähige Datenbank auf Basis von PostgreSQL
- Installation von Python ≥ 3.8
- Lauffähige Installation (Buildout) des Web CMS Plone in der Version $\geq 5.2.4$

1.3 Installation von edi.substanceforms für das Web-CMS Plone

Git Clone von edi.substanceforms in das "src" Verzeichnis der Plone-Installation

```
git clone https://github.com/educorvi/edi.substanceforms.git
```

Einträge in der Datei buildout.cfg der Plone-Installation:

```
eggs:
    ...
    edi.substanceforms

develop:
    ...
    src/edi.substanceforms
```

Durchführung eines neuen Buildouts im Wurzelverzeichnis der Plone-Installation:

```
./bin buildout
```

Installation des neuen Produkts im Plone Controlpanel

2. Ausgangssituation

Die Berufsgenossenschaft Energie Textil Elektro Medienerzeugnisse (BG ETEM) pflegt für die Betriebe der Branche Druck und Papierverarbeitung eine Online-Datenbank mit speziellen Chemikalien und Chemikaliengemischen, die in den Produktionsprozessen der Branche Druck- und Papierverarbeitung zum Einsatz kommen. Zielsetzung des Bereiches Prävention der BG ETEM ist es, über diese Online-Datenbank auf die in den Chemikalien enthaltenen Gefahrstoffe hinzuweisen, und die Unternehmerinnen, Unternehmer und Arbeitsschützer zu motivieren, vor allem die Chemikalienprodukte einzusetzen, die mit möglichst geringen Gesundheitsgefahren für die Beschäftigten der Branche verbunden sind.

So ist beispielsweise bei Chemikaliengemischen darauf zu achten, dass ein möglichst hoher Flammpunkt für einen geringen Eintrag von giftigen Dämpfen in die Atemluft steht und dass Druckbestäubungspuder verwendet werden, die eine bestimmte Mindest-,Körnungsgröße aufweisen, um nicht lungengängig zu sein.

Die Online-Datenbank trägt deshalb den Namen **Emissionsarme Produkte**.

Die Datenbank wurde auf Basis des Web-Content-Management-Systems Plone entwickelt. Die Eigenschaften der Chemikalienprodukte, deren Inhaltsstoffe, Hersteller und vieles mehr wurden bislang in einer objektorientierten Datenbank. Die redaktionelle Arbeit zur Bearbeitung und Pflege sowie auch die Recherche der Endbenutzer erfolgte über die Leistungsmerkmale des CMS Plone.

Sowohl die Oberflächen für die Benutzer, Autoren und Redakteure als auch die Datenstrukturen in dieser Datenbank sind veraltet. Die Möglichkeit, Chemikalien gezielt nach gewissen Parametern zu durchsuchen, ist aufgrund der technischen Eigenschaften der objektorientierten Datenbank nur sehr eingeschränkt.

Zu den Chemikalienprodukten gehören:

- Wasch- und Reinigungsmittel für den Offsetdruck
- Sonderreiniger für die manuelle Anwendung im Offsetdruck
- Reinigungsmittel im Etikettendruck
- Staubarme Druckbestäubungspuder

3. Zielsetzung

Die Zielsetzung des Projekts besteht in der grundsätzlichen Anpassung und Modernisierung der zugrunde liegenden Softwarearchitektur. Vor allem die Umstellung von einer objektorientierten Datenbank auf eine relationale Datenbank bildet eine wesentliche Zielsetzung des Projekts. Daneben soll für die Endbenutzer eine verbesserte und vor allem schnellere Recherche über die in der Datenbank gespeicherten Chemikalienprodukte ermöglicht werden. Die redaktionelle Bearbeitung der Daten soll weiterhin über das bestehende Web-Content-Management-System Plone ermöglicht werden.

Im Rahmen eines Vorprojektes wurde ein ERM-Modell für eine relationale Datenbank entwickelt und anhand von Test-Datensätzen erprobt. Außerdem wurden im Rahmen des Vorprojekts die technischen Möglichkeiten zur Migration der Daten aus der bestehenden objektorientierten Datenbank in die relationale Datenbank getestet. Die im Vorprojekt erzielten Ergebnisse wurden als Proof of Concept gesichert. Dementsprechend konnte in diesem Projekt auf die Ergebnisse des Vorprojekts sowohl in Form von Softwarebestandteilen als auch in Form von Dokumentation zurückgegriffen werden.

4. Anforderungsanalyse

Die Anforderungsanalyse ist für die hier vorliegende technische Dokumentation vor allem vor dem Hintergrund des Verständnisses für die technischen Zusammenhänge der realisierten Softwarelösung von Bedeutung. Dieses Kapitel ist deshalb als Gegenüberstellung der Kundenanforderungen mit den im Projekt entwickelten Artefakten zu verstehen.

4.1 Bedienbarkeit über das Web-CMS Plone

Die folgenden Anforderungen werden durch die im Projekt entwickelten Nutzerformulare und Einzelansichten erfüllt:

- Formulare zum Anlegen von Herstellern und Chemikalienprodukten durch die Autoren und Redakteure mit integrierter Plausibilitätsprüfung und Speicherung im RDBMS
- Einzelansichten für Hersteller und Chemikalienprodukte und Auflistung von deren Eigenschaften für den redaktionellen Prozess
- Möglichkeit zur Löschung von einzelnen Datenbankobjekten aus den Einzelansichten
- Formulare zur Aktualisierung von Herstellern und Chemikalienprodukten
- Suche nach Produkten und Herstellern im Rahmen des redaktionellen Prozesses (keine Endbenutzersuche)

Außerdem waren die folgenden Punkte zu berücksichtigen und wurden entsprechend umgesetzt:

- Nutzung der Benutzerverwaltung des Web-CMS Plone und des Rechte- und Rollenkonzeptes von Plone
- Entwicklung der CMS-Artikeltypen "Datenbank" und "Tabelle". Die Inhaltstypen erlauben den Autoren und Redakteuren den freien Aufbau einer Navigationsstruktur innerhalb des CMS, stellen die technische Verbindung zu den Datenbankobjekten her und erlauben die Nutzung des Berechtigungskonzeptes von Plone bis auf die Ebene der Datenbanktabellen.
- Entwicklung von Methoden zur expliziten Prüfung der Rollen von Benutzern im Kontext der Formulare mit denen einzelne Datensätze bearbeitet werden können:
- **userCanEdit**, prüft das Schreibrecht von Benutzern
- **userCanReview**, prüft das Recht von Benutzern zur Veröffentlichung

Für die Artikeltypen Datenbank und Tabelle verwenden wir einen eigenen Workflow, der die Berechtigungen für die Status wie folgt definiert:

- **privat**: Lese- und Schreibrechte für die CMS-Rollen Owner und Manager
- **Entwurf**: Lese- und Schreibrechte für die CMS-Rollen Editor und Manager, Leserecht für Reviewer
- **zur Veröffentlichung eingereicht**: Lese- und Schreibrechte für die CMS-Rollen Reviewer und Manager, Leserecht für Editor
- **Veröffentlicht**: Lese- und Schreibrechte für die CMS-Rollen Reviewer und Manager, Leserecht für Anonymous

4.2 Responsive Webdesign

Auch hier ist das CMS Plone hilfreich. Plone selbst unterstützt von Haus aus Responsive Webdesign. Das Webdesign des Kunden verwendet außerdem Bootstrap als CSS- und Javascript- Basis. Dementsprechend wurde für die Realisierung aller HTML-Templates im Projekt der HTML-Markup des [Bootstrap Frameworks in der Version 4.6](#) verwendet.

Bootstrap teilt jeden Bildschirm in ein Raster aus 12 Spalten ein und ordnet den Endgeräten je nach Größe des tatsächlichen physikalischen Bildschirms eine von 4 Bildschirmgrößen:

- xs (extra-small),
- sm (small),
- md (medium) und
- lg (large)

zu.

In den Templates des Plone-Add-Ons wird definiert, welche Template-Elemente bei welcher Bildschirmgröße wie viele Spalten einnehmen sollen. So lässt sich beispielsweise an einem Arbeitsplatzcomputer ein Bild neben einer Tabelle anzeigen, während es am Smartphone nicht möglich ist und deshalb automatisch (responsive) unter der Tabelle angezeigt wird.

4.3 Barrierefreiheit gemäß BITV 2.0 (WCAG 2.1 Level AA)

TODO: Bildhandling überarbeiten! Bei korrekter Verwendung des HTML-Markups von Bootstrap ist die Barrierefreiheit aus technischer Sicht gewährleistet. Ausserdem wird in den Formularen zum Upload und zum Update der Bilder zu Herstellern und Chemikalienprodukten sichergestellt, dass Texte für Titel, und Bildbeschreibung angegeben werden können. Die Bilder selbst werden als Artikel im CMS Plone gespeichert. In den Einzelsichten für Hersteller und Chemikalienprodukte werden die Titel- und Beschreibungstexte in den \ Tags verwendet:

- title = Titel des Bildes
- alt = Kurzbeschreibung des Bildes (außerdem Verwendung in der Bildunterschrift)

4.4 Integrität der gespeicherten Daten

TODO: Datenbankdesign erwähnen? - ja! Die Integrität der gespeicherten Daten wird durch Datentypisierung und fachliche Plausibilitätsprüfungen sichergestellt. Somit wird jeder von den Autoren und Redakteuren eingetragene Wert sowohl im Hinblick auf das Datenformat als auch im Hinblick auf die fachliche Richtigkeit hin geprüft.

5. Bestandteile der Applikation (Module und Klassen)

Alle entwickelten Bestandteile des Projekts befinden sich im Plone-Add-On **edi.substanceforms**. Dieses Add-on beinhaltet beinhaltet Artikeltypen Content-Types (Artikeltypen), mit denen in Plone die Inhalte der Seiten gepflegt werden, sowie alle benötigten Formulare und Ansichten (Forms und Views), für das Management der Chemikalienprodukte mit dem CMS Plone. Die Bestandteile der Applikation werden in den folgenden Abschnitten detaillierter beschrieben.

Für die Erstellung der Plone-Inhaltstypen (Content-Types, Viewklassen, Viewtemplates...) kommt der Generator **mrbob** zum Einsatz. Dieser ermöglicht den Fokus auf die essentielle Programmierung und Anpassung der Inhaltstypen durch massive Zeitersparnisse gegenüber des manuellen Anlegens.

5.1 Artikeltypen für die Pflege von Inhalten

Die Content-Types haben für die Autoren und Redakteure eine besondere Bedeutung. Mit deren Hilfe werden in der Struktur der Plone-Site Inhalte angelegt um diese für Benutzerinnen und Benutzer zu präsentieren.

Daneben können in den Content-Types Daten gespeichert werden, die für die Steuerung der Anwendung benötigt werden. Das Package `edi.substanceforms` beinhaltet folgende Content-Types:

5.1.1 Datenbank

Die Datenbank **Emissionsarme Produkte** kann mit diesem Artikeltyp an beliebiger Stelle in der Inhaltsstruktur der Website eingebunden werden. Die Autoren und Redakteure können mit entsprechenden Feldern die Datenbank beschreiben und für die Benutzerinnen und Benutzer präsentieren. Außerdem stellt die Datenbank die technische Verbindung zur physikalischen Datenbank her. D.h. es werden mit diesem Artikeltyp alle notwendigen Parameter für die Verbindung zur Datenbank gespeichert.

Aus der Sicht des Web-CMS-Plone stellt die Datenbank einen Ordertyp dar in dem Objekte vom Content-Type Tabelle gespeichert werden.

5.1.2 Tabelle

Der Content-Typ Tabelle stellt die technische Verbindung zu den Tabellen der physikalischen Datenbank her. Bereits beim Anlegen des Content-Types in Plone wird der Autor und Redakteur gefragt, welche der Tabellen der Datenbank angezeigt werden soll. Mit dem Content-Typ Tabelle wird die Absicht verfolgt im Context einer Tabelle die darin gespeicherten Datensätze anzuzeigen als wären sie selbst als Objekte im CMS Plone gespeichert.

Aus der Sicht des Web-CMS Plone stellt die Datenbank ebenfalls einen Ordertyp dar, in dem Dateiobjekte (Bilder, PDF-Dateien) gespeichert werden, die für die Präsentation der Datensätze, also z.B. Hersteller, Chemikalienprodukte benötigt werden. Für die Speicherung dieser Objekte wird ganz bewusst die objektorientierte Datenbank genutzt. In den Datensätzen werden Referenzen zu den Dateiobjekten gehalten.

5.2 Formulare

Formulare werden nach dem CRUD-Begriff (**C**reate, **U**ppdate, **D**eleate) für folgende Anwendungsfälle benötigt:

Hinzufügen von Datensätzen in die Datenbank (create) Aktualisierung von Datensätzen in der Datenbank (update)

Das Löschen von Datensätzen wird mit einem funktionalen View realisiert.

Ausserdem wird ein Formular für die Suche von Datensätzen in der Datenbank benötigt.

5.3 Views

5.3.1 Einzelansicht für die Datenbank

In der Einzelansicht für die Datenbank (`datanbank_view.pt`) werden vorher zum CMS hinzugefügte Tabellen der Datenbank angezeigt. Eine Tabelle repräsentiert ihrerseits Hersteller oder Chemikalienprodukte.

5.3.2 Einzelansicht für die Tabelle

Die Einzelansicht für die Tabelle (`tabelle_view.pt`) dient dem Listing aller Datensätze in der Tabelle und enthält entsprechende Suchparameter, nach denen die Trefferliste bei Bedarf gefiltert werden kann. So ist beispielsweise der Ausschluss aller Gefahrstoffgemische möglich, die nicht als Sonderreiniger definiert sind.

5.3.3 Einzelansicht für Hersteller und Gefahrstoffprodukte

Für jede Art von Produkten gibt es ein individuelles Viewtemplate. In diesem Viewtemplate wird eine Tabelle erzeugt, in der die entsprechenden Eigenschaften des Produktes zur Anzeige gebracht werden. Außerdem erfolgt hier die Darstellung der gespeicherten Bilder. Folgende Viewtemplates sind vorhanden:

- `substance_view.pt`
- `substance_mixture_view.pt`
- `spray_powder_view.pt`
- `manufacturer_view.pt`

Viele in der Datenbank gespeicherte Werte sind nicht lesbar. Daher erfolgt in der Viewklasse `single_view.py`, welche allen Einzelansichten zugrunde liegt, die Key-Value-Zuweisung und die Übersetzung von Werten.

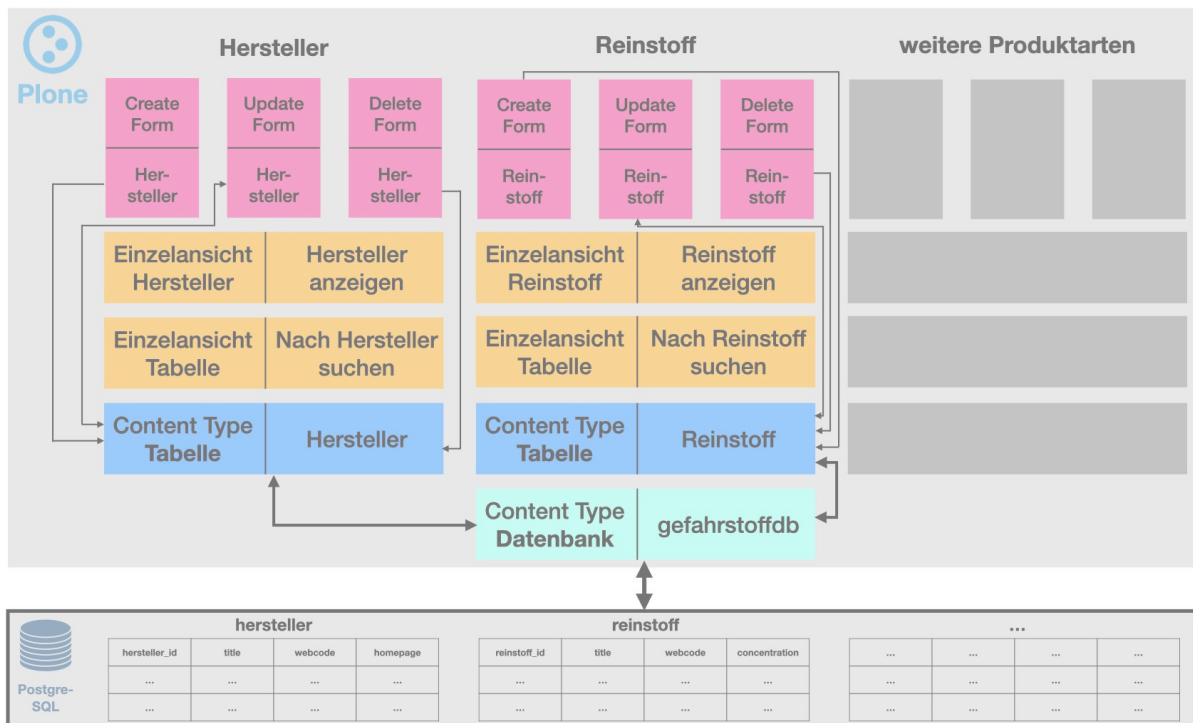
5.4 Funktionale Views

Ein funktionaler View zeichnet sich dadurch aus, entweder kein eigenes Template zu besitzen, oder eines das auf die Bestätigung einer Aktion beschränkt ist. Zu den funktionalen Views gehören:

- Delete View
- single-View
- ...

So besitzt beispielsweise der Delete View ein Template zur Bestätigung des Löschens des Artikels per Auswahl einer Checkbox, während in den single-view alle Methoden der Einzelansichten ausgelagert wurden.

6. Aufbau des Software-Packages



[Bild öffnen](#)

7. Automatische Testverfahren

Die Entwicklung von `edi.substanceforms` wurde im grundsätzlichen Ansatz testgetrieben durchgeführt. Wie im Kapitel "Bestandteile" beschrieben, wurden die Inhaltstypen **Datenbank** und **Tabelle** sowie alle Ansichten (Views) und Formulare (Forms) wurden zunächst mit dem Generator `mr.bob` generiert.

Im Zuge der Generierung werden bereits Templates für funktionale Tests im Package angelegt. Ausserdem werden in der Datei `setup.py` des Packages die Abhängigkeiten zur Durchführung automatisierter Tests vermerkt. Um die automatisierten Tests ausführen zu können müssen diese Abhängigkeiten zunächst installiert werden. Dazu muss folgende Änderung in der `buildout.cfg` vorgenommen werden:

```
eggs =
...
edi.substanceforms [test]
```

Danach muss der Buildout mit dem Befehl `./bin/buildout` im Wurzelverzeichnis der Installation erneut durchgeführt werden.

7.1 Scope der Funktionstests

Funktionstests im Web-CMS Plone verfügen über eine echte (objektorientierte) Datenbank und eine Komponentenarchitektur. Darüber hinaus kann mit Python-Code ein Browser simuliert werden. Wenn mit dem Python-Browser auf eine Seite zugegriffen wird, ist die komplette Transaktionsmaschinerie von Plone im Einsatz. Damit dies funktioniert, verpackt die Testschicht die Datenbank in einen Demospeicher, der einen regulären Speicher kapselt. Wird von den Tests etwas in die Datenbank geschrieben, speichert der Demostorage es in temporären Feldern des Arbeitsspeichers. Nach jedem Test wird der Demostorage gelöscht. Damit sollten die funktionalen Tests fast so schnell ablaufen wie Integrationstests oder gar Unit-Tests und das trotz des zusätzliche Overheads, der durch das Durchlaufen der Plone-Transaktionsmaschine aufgebaut wird.

Zu beachten ist dabei, dass der simulierte Browser reinen Python-Code darstellt. Javascript-Code kann somit im Rahmen der funktionalen Tests nicht getestet werden.

Ebenso wurden die Datenbank-Transaktionen zur relationalen Datenbank auf Basis von PostgreSQL nicht in die Funktionstests einbezogen. Die tatsächliche Kundenkonfiguration wird erheblich von einer Test- oder Entwicklungskonfiguration abweichen. Die generische Implementierung der notwendigen technischen Voraussetzungen für die Installation und Konfiguration der Testumgebung ist nicht möglich.

Die in `edi.substanceforms` implementierten Funktionstests umfassen die folgenden Themen:

- fehlerfreie Installation des AddOns im Content-Management-System Plone
- Anlegen der Inhaltstypen Datenbank und Tabelle
- Assoziation der View- und Formklassen zu den Inhaltstypen

7.2 Test-Verzeichnis im Package

Alle für die Ausführung der Funktionstests mittels Testroboter (`(plone.app.testrobot)[https://pypi.org/project/plone.app.robotframework/]`) notwendigen Konfigurationsdateien sowie die Dateien mit den Funktionstests befinden sich im Verzeichnis `../src/edi/substanceforms/tests/..`:

```
...src/edi/substanceforms/tests$ ls -la
total 132
drwxrwxr-x 4 plone_buildout plone_buildout 4096 Dec  4 13:03 .
drwxrwxr-x 9 plone_buildout plone_buildout 4096 Dec  4 12:28 ..
-rw-rw-r-- 1 plone_buildout plone_buildout    0 Nov 25 10:50 __init__.py
drwxrwxr-x 2 plone_buildout plone_buildout 4096 Dec  4 10:45 __pycache__
drwxrwxr-x 2 plone_buildout plone_buildout 4096 Nov 25 10:50 robot
-rw-rw-r-- 1 plone_buildout plone_buildout 3042 Nov 25 10:50 test_ct_datenbank.py
...
-rw-rw-r-- 1 plone_buildout plone_buildout 1564 Dec  4 11:33 test_view_update_mixture-form.py
-rw-rw-r-- 1 plone_buildout plone_buildout 1539 Dec  4 11:33 test_view_update_powder_form.py
-rw-rw-r-- 1 plone_buildout plone_buildout 1570 Dec  4 11:33 test_view_update_substance-form.py
-rw-rw-r-- 1 plone_buildout plone_buildout 1518 Dec  4 11:33 test_view_update_view.py
```

7.3 Ausführung der Tests

Die Tests für `edi.substanceforms` können aus dem Buildout-Verzeichnis heraus mit dem folgenden Befehl ausgeführt werden:

```
$ ./bin/test edi.substanceforms

...
Ran 65 tests with 0 failures, 0 errors, 0 skipped in 3.067 seconds.
Tearing down left over layers:
Tear down edi.substanceforms.testing.EdiSubstanceformsLayer:IntegrationTesting in 0.000 seconds.
Tear down edi.substanceforms.testing.EdiSubstanceformsLayer in 0.004 seconds.
Tear down plone.app.contenttypes.testing.PloneAppContenttypes in 0.020 seconds.
Tear down plone.app.event.testing.PAEventLayer in 0.004 seconds.
Tear down plone.app.testing.layers.PloneFixture in 0.028 seconds.
Tear down plone.testing.zope.Startup in 0.003 seconds.
Tear down plone.testing.zca.LayerCleanup in 0.001 seconds.
```

8. Installation und Konfiguration im CMS Plone

Das Package **edi.substanceforms** wird auf der Plattform GitHub gehostet.

Um edi.substanceforms im Web-Content-Management-System Plone zu installieren, wird das Package zunächst via git clone in das Sourcen-Verzeichns (src) der Buildout-Installation heruntergeladen und für die Installation in der Datei **buildout.cfg** eingetragen. Die Installation erfolgt dann im Rahmen des Buildout-Prozesses

```
{buildout-home}$ cd src
{buildout-home}/src$ git clone https://github.com/educorvi/edi.substanceforms.git
```

Einträge in der Datei buildout.cfg der Plone-Installation:

```
eggs:
    ...
    edi.substanceforms

develop:
    ...
    src/edi.substanceforms
```

Durchführung eines neuen Buildouts im Wurzelverzeichnis der Plone-Installation:

```
{buildout-home}$ ./bin buildout
```

Anschließend kann der Plone-Client hochgefahren werden, dies geschieht mit dem Befehl:

```
{buildout-home}$ ./bin/{instance} start
```

Danach muss das Package in den Plone-Einstellungen installiert werden.

--> Einstellungen ({url}/{@@plone_control_panel}) --> edi.substanceforms --> **Installieren**

Das Package wurde nun erfolgreich installiert und ist einsatzbereit.

Nun kann mit dem Hinzufügen einer Datenbank begonnen werden.

9. Lizenzvereinbarung für edi.substanceforms

Für jedes Softwareprodukt ist es von essentieller Bedeutung, Rechte und Ansprüche auf das geistige Eigentum zu definieren und zu schützen. Hierfür werden meist Lizenzen verwendet, die gewisse Rechte klar festlegen. In unserer Firma folgen wir dem Open Source Gedanken und stellen unsere Software der Community frei zur Verfügung. Im Gegenzug bedienen wir uns an anderen Open Source Projekten, die uns als Grundlage für unsere Software dienen.

So kommt für die gesamte Software und die Dokumentation die [MIT-Lizenz](#) zum Einsatz. Für die MIT Lizenz habe ich mich entschieden, da sie Business- und Open Source freundlich ist, die Verwendung durch Andere ermöglicht ohne deren gesamten proprietären Quellcode aufzudecken zu müssen (eine Nennung in der Lizenzangabe genügt), gleichzeitig jedoch im Gegensatz zu GPL weiterhin eine Monetarisierung möglich macht.

Die Dokumentation in gedruckter Form, sprich der PDF Export, ist unter [CC BY 4.0](#) lizenziert. Diese Lizenz habe ich gewählt, da Sie die Verwendung, Veränderung und Veröffentlichung durch Andere (sogar für die kommerzielle Nutzung) ermöglicht, jedoch eine Nennung des Urhebers voraussetzt.

10. Anhang

10.1 Test 1 2 3 4 5
