Document generator

I passed a little time to produce a document generator named *sara* using *jinja2* templating

Quickstart

- Clone project from https://gitlab.euclid-sgs.uk/wg_sara/sara-documentation-composer.git
- Create virtual env (or let pyCharm or you tool creating it)

You have to use **two** folders

- the standard folder where skelettons are located,
- the custom folder where custom modifications

Run the main command with

```
python sara/cmd.py render --location=sara\templates --location=tmp\custom_templates --project=tmp/
project.yml --document=tmp/document.yaml --template=software_review_master.adoc
```

It will produce a *asciidoc* document based upon *templates/software_review_master.adoc* itself based upon *templates/master.adoc*

```
= {% block title %}{{doc.title}}{% endblock %}

{% include 'fragments/doc_identification_cartouche.adoc' %}

{% include 'fragments/doc_lifecycle_cartouche.adoc' %}

{% include 'fragments/doc_issues_cartouche.adoc' %}

{% block body %}body{% endblock %}
```

Each include use a fragment, a reusable part of document

Fragment called *fragments/doc_issues_cartouche.adoc* will produces an array from data

```
//
// Documentation lifecycle cartouche
//

[cols="1,1,1,3,4",stripes="none"]]
|=============================================
| *Issue* | *Date* | *Page* | *Description of Change* | *Comment*
//{% for issue in doc.issues %}
| {{issue.id}} | {{issue.date}} | {{','.join(issue.page)}} | {{issue.description}} | {{issue.comme
nt}}
//{% endfor %}
|=============================================
```

You could focus on the "for in loop", that iterates over a list of issues associated to document.

Issues are stored as part of a document definition :

```
title: document title
```

```
version: '1.0'
date: 2019-01-30
reference: REC-FOO-A
custodian: mister.custodian
#...
issues:
  - id: '0.1'
    description: Typo
    date: '2018-11-01'
    comment: file to view the source code until proper documentation is generated.
    page:
      - '2'
  - id: '0.2'
    description: Typo very long
    date: '2018-11-01'
    comment: balbalbalbalbalb
    page:
    - '2'
  - id: '0.3'
    description: Typo
    date: '2018-11-01'
    comment: datetime field formatted using formatter.
    page:
    - '2'
    - '3'
```

The asciidoc generated text will be

```
[cols="1,1,1,3,4",stripes="none"]]
|=============================================
| *Issue* | *Date* | *Page* | *Description of Change* | *Comment*
//
| 0.1 | 2018-11-01 | 2 | Typo | file to view the source code until proper documentation is generat
ed.
//
| 0.2 | 2018-11-01 | 2 | Typo very long | balbalbalbalbalb
//
| 0.3 | 2018-11-01 | 2,3 | Typo | datetime field formatted using formatter.
//
|=============================================
```

This renders as

Screenshot_2019-02-21%20AsciidocFX%20Editor.png

Files

Screenshot_2019-02-21 AsciidocFX Editor.png	16.3 KB	2019-02-21	Marc DEXET