**Solutions Strategy Guide**

Contents

[About 2](#_Toc452408530)

[Guiding Examples 3](#_Toc452408531)

[Pros and Cons 4](#_Toc452408532)

[Possible difficulties 4](#_Toc452408533)

[My Opinion 4](#_Toc452408534)

[Amount of work expected 4](#_Toc452408535)

[Forges 4](#_Toc452408536)

[Pros and Cons 5](#_Toc452408537)

[Possible difficulties 5](#_Toc452408538)

[My Opinion 5](#_Toc452408539)

[Amount of work expected 5](#_Toc452408540)

[Wiki 5](#_Toc452408541)

[Pros and Cons 6](#_Toc452408542)

[Possible difficulties 6](#_Toc452408543)

[My Opinion 6](#_Toc452408544)

[Amount of work expected 6](#_Toc452408545)

[Content Management Systems 6](#_Toc452408546)

[Pros and Cons 7](#_Toc452408547)

[Possible difficulties 7](#_Toc452408548)

[My Opinion 7](#_Toc452408549)

[Amount of work expected 7](#_Toc452408550)

[Building From Scratch 7](#_Toc452408551)

[Pros and Cons 8](#_Toc452408552)

[Possible difficulties 8](#_Toc452408553)

[My Opinion 8](#_Toc452408554)

[Amount of work expected 8](#_Toc452408555)

# About

We gathered a list of existing technologies to refer to and/or possibly use for this project, however the list ended up being pretty large so we divided this project into five different approaches: using existing examples and adding additional features we could build ourselves (Ruby Gems, PyPI, etc), using forges like gitlab, github, gogs, and doing something like an organization on github, a “wiki” style like CSPOGIL, through the use of a content management system like Drupal, Orchard Project, etc, or building from scratch.

# Guiding Examples

Guiding examples are existing projects that do what we want or COULD do what we want with a few modifications. This would likely require the least amount of work, some plugins could be built, maybe we’d need access to an API which some of these do provide, like Atom.io and Ruby Gems.

## Pros and Cons

|  |  |
| --- | --- |
| Pros | Cons |
| Least Amount of work | May not tailor to exactly what we’re looking for |
| Ruby Gems has an existing package manager | Not all of these websites are open source |
|  | Possibly 3rd party hosting/ someone needs to oversee the project / monitor activity |
|  | May need extra plugins to achieve what we want |
| Possibly learning a new framework or language | May take hours to analyze and understand code that was written if there are a lack of comments |

## Possible difficulties

* We may need to build some applications for extended functionality
* Sifting through code to understand what is going on, if we do reuse a site and add to it

## My Opinion

I’d say that this approach is going to take the least amount of resources, mainly because everything is essentially built for us. It’s just a matter of analyzing the code and understand what is going on at each step. Though none of these actually cover the full functionality of what we’re looking for, so we’re going to need to consider building features on top of what’s given. Then we need to consider how these applications/sites were built, could they even support the specifications we desire.

### Amount of work expected

If we were to go with this approach, the project could be done in a month or two. It’s a somewhat broad range, but the main problem is finding a way to use the sites/applications to fit our specifications, and for that we’re going to have to add to what already exists.

# Forges

Forges are essentially existing project/repository managers like github, gogs, gitlab. Etc. Some of these “forges” provide features like github’s organization feature, which would allow for the monitoring of multiple repositories.

## Pros and Cons

|  |  |
| --- | --- |
| Pros | Cons |
| Not as much work required as building from scratch | Needs a person or a group of people to oversee the “organization” |
|  |  |

­­­­­­

## Possible difficulties

## My Opinion

### Amount of work expected

# Wiki

Wikis are sites or tools that support the collaborative editing of its content.

## Pros and Cons

|  |  |
| --- | --- |
| Pros | Cons |
| Instant collaboration | Text heavy? |
| Well kept page revision | Contingent on users adding appropriate content and reporting what’s inappropriate |
|  | Some person or entity needs to monitor whats being added |

## Possible difficulties

## My Opinion

### Amount of work expected

# Content Management Systems

## Pros and Cons

|  |  |
| --- | --- |
| Pros | Cons |
| Typically support version control |  |
| Indexing |  |

## Possible difficulties

## My Opinion

### Amount of work expected

# 

# Building From Scratch

## Pros and Cons

## Possible difficulties

## My Opinion

### Amount of work expected