Priority color scheme: Green (inconsequential) < Yellow < Orange < Red (highly important)

Parameters	Remarks	Preferred technology/framework	Priority
SSGs			
Language - Javascript	Most of the people in our community write in Javascript. It will be significantly easier and more interesting if developers can use Infusion in tandem with the static site generator, rather than to have to pick up a new programming language such as Python or Go and new tools/frameworks as well.	Gatsby, 11ty	
Consistency across different projects	Several of our recent websites are being built using 11ty and deployed on Netlify. It's still a new tool for us, but we've got a growing number of developers who have experience with it (more so than Hugo or Jekyll or the others)	11ty	
CMS			
Easy of adding new content for developers/Content Management System	Netlify CMS for IDRC site, Headless Wordpress for We Count site. This allows for a more polished WYSIWYG experience for non-technical site editors, we want to choose a tool that we	Content dependent. Netlify CMS for static ones and headless WordPress for dynamic functionality ones.	

	can use for lots of different kinds of sites, and build a body of knowledge and facility with it across projects.		

Existing Fluid Project Repos:

Name	**SSG Used**
Infusion Docs	Docpad
So just repair it	Hugo
Floe Project	Hard-Coded
Fluid Project	Docpad
Infusion Nexus Demos	Hardcoded
Docs Inclusive Learning	Docpad
Docs-template	Docpad
guide.inclusivedesign.ca	Docpad
http://eco-op.inclusivedesign.ca/	Eleventy
website-cities	Hugo
website-nide	Hard-Coded
inclusivedesign.ca	Wordpress

Static Site Generators Comparison:

	Gatsby	HUGO	jekyll	11ty
Github Stars	42.7k 🌟	42.2k 🌟	39.9k🌟	4.6k🌟
Language	JavaScript	Go	Ruby	Javascript
Templates	React	Go	Liquid	Multiple support
Integration	React, Webpack, GraphQL	Markdown, GitLab Pages, Buddy	Gitlab Pages, Comment It, Github Pages	GH-pages, Markdown
Pros	Progressive Web Application is developed.	Faster to develop.	Built-in Development Server.	Supports multiple template languages.
	Supports GraphQL	Don't need time to figure out configurations.	SEO-conscious in nature.	Zero boilerplate client-side JavaScript.
	Supports a huge ecosystem of Plugins.	Cross-Platform Availability	Good plugin and theme support base.	Zero Config: Works with the project's default file structure.
	Well written tutorial and documentation	Supports TOML, YAML, and JSON for the front matter and strong theme base	Liquid templating is convenient and easy to understand.	Written in JS, we can write our own plugins and data processing tools as per need.
	Vast and active community. Help easily available.	Enterprise-ready. With support for multilingual sites	Readily integrable with Github pages.	zero-config by default but flexible conf. Options
	Gatsby	HUGO	jekyll	11ty

Cons	Needs a strong understanding of ReactJS.	No Plugin support	No Image editor	Small community. Help isn't easily available.
	Beginners find it tough to develop.	Uses Go's Template instead of Liquid which is non-user friendly.	Build time increases with the addition of plugins	Small Plugin support base. Only three official plugins are present.
	Build time increases dramatically with files and styles	No asset pipeline.	Slower to build compared to Hugo.	Build times are slow compared to Hugo and Jekyll.
	Everything has to be stored in memory and hence RAM tends to run out while building it.	No XML Support for data feeds.	Restricted file structure. Iterating through data files is limited in nature.	The template base isn't strong, unlike Jekyll and Hugo.
Recommendation	Not suitable as per our needs.	Candidate for further evaluation.	Candidate for further evaluation.	Too small/poorly adopted to consider.

Detailed Review of SSGs:

Each Static Site Generator has its own pros and cons. The right choice is based on what technologies and frameworks the present fluid community are familiar with, what technologies are presently being used in other projects to maintain consistency. Moreover build speed considerations, community support, ease of understanding for new contributors, ordered file structure, plugin support base, and the needs of the website also play an important role in deciding the static site generator.

Deciding Points:

- ***Gatsby***
    1) It is framework dependent.
    2) We currently do not need GraphQL and not everyone is well-rehearsed with React
    3) Gatsby requires a good understanding of React for development. The closer we get to "apps" on the "content site to app spectrum" the more we'd want to consider one of the framework-based tools. Our project sites are though more "content-driven" in nature.

4) Good plugin base and well documentation are general factors that can be found in other SSGs too. Build times increase exponentially as files increase and everything has to be stored in RAM which tends to run out as reported by developers.
5) Each plugin that we use on a Gatsby site adds load to the site's overall performance. Sometimes we can fix the situation by properly configuring those plugins, but quite often we just have to limit their number for a site thus imposing a restriction.

- ***Hugo***
    1) It's easy to install software. Hugo doesn't depend on administrative privileges, databases, runtimes, interpreters or external libraries.
    2) Sites built with Hugo can be deployed on S3, Github Pages, Dropbox or any web host. Hugo is fast & powerful. It's written for speed and performance.
    3) Great care has been taken to ensure that Hugo's build-time is as short as possible. It takes milliseconds to build an entire site for most setups. Hugo is flexible and designed to work how we do. We can organize our content however we want with any URL structure.
    4) We can declare our own content types and define our own metadata in YAML, TOML or JSON.
    5) Documentation and community support are sufficient enough to complete the project. Versioning can be realized by the usage of tags. There is an inbuilt provision for tag-specific pages.

- ***11ty***
    1) It provides a flexible file structure system thus we can migrate while we maintain our current file structure.
    2) It supports multiple file extensions and not just markdown. It's written in JS which is not going to lose support any time soon.
    3) But it has its downsides which can't be ignored. It has a small community, and thus help isn't easily available. It provides a flexible directory structure, every page needs to have a front matter that specifies the template it needs to use. This process is automated in other SSGs like Hugo.
    4) Versioning can be done by collections but it's not as intuitive as HUGO tags i.e. no tag-specific pages.
    5) Non-availability of enough documentation and less support from the community can act as a huge factor affecting the development of our project.

6) It has a meager plugin base consisting of only three official plugins. Plugins, if needed, need to be self-written which will increase the time required to complete the project, thus diverting attention from other important topics.
7) The positive points that 11ty puts forward like the use of multiple templating language support aren't of much use to our project. Build times are slow and the template engine base isn't strong.

- ***Jekyll***
    1) It is more like a file-based CMS, without all the complexity. Jekyll takes your content, renders Markdown and Liquid templates, and spits out a complete, static website ready to be served by Apache, Nginx or another web server.
    2) Jekyll is the engine behind GitHub Pages, which you can use to host sites right from your GitHub repositories. Compatibility with gh-pages is the major reason why most of the developers choose Jekyll over Hugo. Fluid project and FLOE Project websites are to be hosted on private servers though and hence the factor of gh-pages integration becomes obsolete.
    3) Jekyll has a strict file structure system and iterating over files in the _data folder causes problems due to restricted liquid templating.
    4) For example, site.data.diectory_name will loop through all the files present there without any plugin available to customize the loop. This causes problems if there are other subdirectories in the directory_name folder.
    5) On a brighter side, we can't ignore the fact that Jekyll is readily compatible with the GH-pages. Github can automatically build Jekyll sites.