



UPPSALA
UNIVERSITET

How to Create a (Great) Artifact

Kiko Fernandez-Reyes



Motivation for Artifacts

1. You spend months / years working on a specific problem
2. [After many rejections is] accepted
3. My paper cannot cover all implementation details
4. Wouldn't it be great if I make a tiny bit extra effort for researchers / people wanting to use my research?



UPPSALA
UNIVERSITET

Motivation



Documented, for ease of understanding of my code

Complete, all components from the paper are included

Exercisable, software is usable



Carefully documented and well-structured to the extent that **reuse** is facilitated



Made **permanently** available



Structure of the Artifact

- Folder Structure
- Prerequisites
 - Installation for OSX, Linux, Windows, and VM
- Content
 - Content
 - Deviations From Paper
 - Code Documentation
 - Tests
 - Extra: Linked to paper (on a per section basis)
 - Extra: How to modify your examples
 - Extra: Material that did not fit the paper
 - Extra: Different formats (HTML, PDF, etc)



Folder Structure

Someone downloads your artifact.

Where can I find the...
documentation?
code?
proofs?
examples?

Show and Tell

0. Folder Structure

The folder structure of this artefact is as follows:


```
.  
|--- README.html (Documentation of the artefact in HTML format)  
|--- README.pdf (Documentation of the artefact in PDF format)  
|--- documentation (auto-generated documentation from code)  
|   |--- index.html  
|   |--- ...  
|  
|--- assets  
|   |--- fonts  
|   |--- pandoc.css  
|   |--- submitted-version.pdf  
|  
|--- typechecker-oopl (Type checker)  
|   |--- stack.yaml  
|   |--- LICENSE-MIT  
|   |--- README.md  
|   |--- Setup.hs
```



Prerequisites & Installation

- State your **dependencies**
- **Install them** with the reader
- **Delegate** if there are problems
- Always provide a **VM** in `.ova` format

Either one line or
create a script



[Example](#)

Works with any
virtualisation software





UPPSALA
UNIVERSITET

Content

Briefly explain the library structure

[Example](#)

Content

Deviations from paper

Where is the source documentation?

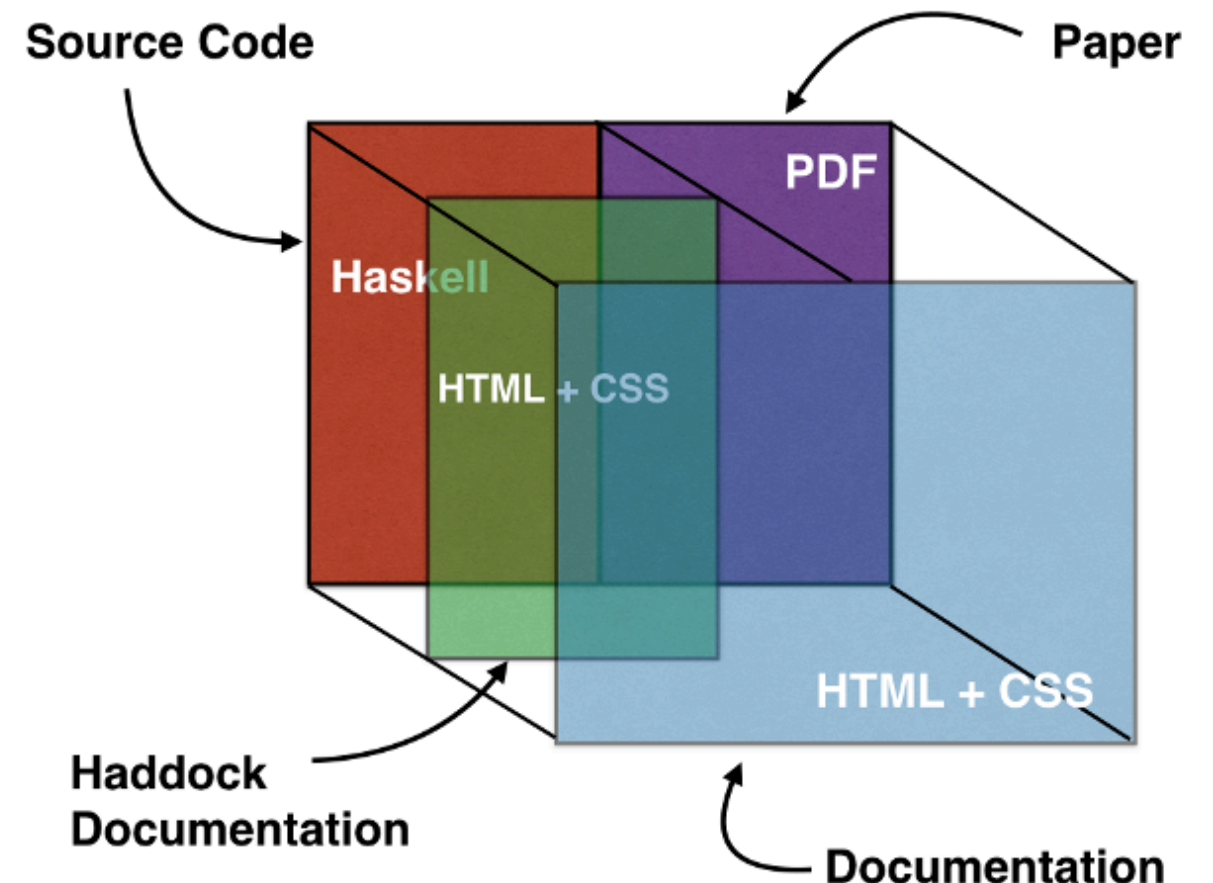
Unit tests



Extras

1. Link to paper!
2. How to **modify** examples, how to **use the library**, and what is the **expected behaviour** or outcome
3. New material if concepts are [too advanced](#)
4. Produce self-contained HTML and other formats, e.g., PDF

Example





UPPSALA
UNIVERSITET

Tools

Make makefiles for overall orchestration of all components

Haddock for generating source code documentation

Pandoc for generating PDF and HTML files from a Markdown file

Vagrant for provisioning virtual machines

Stack for downloading Haskell dependencies, compiling, running tests, etc



Summary

- Make your research...
 - ... useful to the community
 - ... easy to use
 - ... easy to replicate results
- Automate as much as possible
- Total time: 40 h (incremental building, 30 pages)