Code standards guides  
Terms of reference

# Background

Currently every developer has their own way of doing things, which may be fine for small, stand-alone projects but within an enterprise-level project this leads to issues such as:

* **Inconsistent code** in terms of organisation, syntax, formatting, and naming conventions.
* **Code bloat** in production systems, often including code that is either never used or is out-of-date (e.g. hacks for redundant versions of Internet Explorer).
* **Multiple implementations of functionality**, with the consequential maintenance overload due to multiple chunks of code that solve the same problem in different ways, and an inconsistent look-and-feel due to multiple implementations of similar features e.g. tabs, sliders.
* **Bugs and typos**.
* **Uncommented or poorly documented code**.
* **Unreadable code**.
* **Code from one site linked to from other projects** that if updated at source runs the risk of breaking features implemented elsewhere.

While it’s obvious that these inconsistencies and issues can result from **multiple developers** working across **multiple projects**, it can also be true for **solo developers** working on **single projects**. In other words, developers can be inconsistent even with themselves in the same project.

The consequences of this are many:

* Code that is difficult or impossible to maintain.
* Code that is fragile and easy to break in unknown ways.
* Fear of updating or improving code that leads to forked code and/or duplicating functionality using different resources.
* Maintenance overload.
* Slows down development

The reason that we’ve reached this position is primarily that the web presence at St Andrews has evolved organically: there is still no formal, overarching web governance; the University had no central web team until 2006 (13 years after the introduction of the World Wide Web); there are currently no agreed standards; and we have multiple developers within multiple groups (digital communications team, IT Services, Schools and Units, as well as external consultants).

# Objective

Our objectives for introducing code standards and style guides are:

1. Develop an agreed standard for all code and ensure that code within scope adheres to these standards.
2. Use industry best practices.
3. Promote efficiency in maintaning code.
4. Continuously improve our code.

# Benefits

Standard conventions will make it easier for developers to make decisions about how to code which should result in faster development, and make it easier to spot errors and typos.

# Scope

These standards and style guides apply to all new websites and applications for which the digital communications team is responsible (whether developed in-house or outsourced). Guidelines will be written for the following languages:

* HTML
* Cascading Stylesheets (CSS), including pre-processors such as Sass.
* JavaScript, including frameworks such as sjQuery.
* PHP
* XML, including RSS and XSLT.
* Commit messages for version control systems such as Git and Subversion.

These standards and style guides will also apply to existing code in the digital pattern library; these patterns will need to be refactored to comply with these guidelines.

It is understsood that other developers within the University may choose to adopt these guidelines, but at this stage this is not mandatory.

# Constraints

What stops us?

* It's a lot of work to define and document exactly what we want.
* Building consensus with other developers.

How do we enforce it?

* External standards, policies and regulations, e.g.
  + HTML specifications (W3C)
  + CSS specifications (W3C)
  + ECMAScript (JavaScript) specifications
  + PHP specifications
  + XML specifications (W3C), including RSS/Atom and XSLT
  + In-code documentation formatting standards such as PHPDoc, JSDoc, etc.
  + Other relevant code standards such as Microformats (hCalendar, hCard), microdata, FOAF, DOAC, Dublin Core metadata, RDF, WordPress Codex, etc.

It’s imperative that we introduce these code standards and style guides as soon as possible so that we may reduce existing code-debt (code that has been developed prior to the introduction of these standards) in the digital pattern library.

# Assumptions

* We're allowed to enforce governance on front-end code.
* Delivered in a digital manner.
* We will maintain and continuously improve the code standards guides.

# Risks

Not doing

* Current issues (described in Background) and consequences will continue or even get worse.

Doing

* Code that is developed before the standard is completed will need to be refactored.
* May slows down some development.
* Developers refuse to use it because of personal coding preferences and habits.
* Developers may feel frustrated because their coding requirements are not addressed.

# Deliverables

Documentation for each language in scope, delivered in HTML format.

# Stakeholders

Governance

* Chief Information Officer
* Director of Corporate Communications
* Digital communications team

Users

* Digital communications team
* (Development teams)
  + IT Services
  + Schools
  + Units
* (External)
  + Developers
  + Software providers
  + Consultants

# Information (and Data)

* List of applications and scripts for which we are responsible. <https://trello.com/b/dhSK5nzg/service-catalogue>

# Meta

Version 1.3  
Wednesday 30 September 2015

Version 1.2  
Thursday 12 February 2015

Gareth J M Saunders  
Samuel J Parsons