Webonise Lab

Induction Program

Development Conventions

On Web

Coding for S/w vs Web

- While a desktop software runs off the web,
- A web-based software application is intended to run in the web environment.

Difference btw a web server and an application server

Practices

Coding Standards and rules-:

- Never say die.
- Don’t break encapsulation by globalizing variables:
- Lines should have no trailing whitespace at their end.
- All indentation should not use tabs; use 4 spaces instead.

Practices

Variable and method naming convention

- Variables: Use lowercase with underscores to name variables
- Methods: Use CamelCase to name methods

File and Directory Naming Conventions
Folder Structure

- Parent folder name start with Capital letters and use CamelCase but not the child folders
- Linux / UNIX: Reserved Characters And Words Avoid using the following characters from appearing in file names:/,>,<,|,:,&
- All file names are case sensitive.

Good vs Bad

Bad code

We’ll write our code in a pseudocode.

```
FUNCTION comppoly(x)
float y1, y2
float a1=0.1, b1=0.3, a2=2.1, b2=5.3, c=0.22
y1 = a1*x + b1
y2 = a1*x^2 + b2*x + c
return(y2>y1)
END FUNCTION
```

Good code

```
FUNCTION ComparePolynomials(x)
//DECLARE VARIABLES, PARAMETERS
float  y_line, y_quadratic
float  lineParam   = [0.1, 0.3]
float  quadParam = [2.1, 5.3, 0.22]
//CALCULATE THE LINE N QUADRATIC VALUES AT X
y_line              = lineParam[0]*x + lineParam[1]
y_quadratic = quadParam[0]*x^2 + quadParam[1]*x + quadParam[2]

//COMPARE THE FUNCTIONS, RETURNING A LOGICAL
return(y_line > y_quadratic)
END FUNCTION
```

Pairing Up

Pair Programming is a software development technique in which two programmers work together at one work station

Refactor

What?

refactoring is the"... process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure."

Just cleaning up code

- Contrary to idealized development strategy:
- analysis and design
- code
- test

Refactor

Why?

Improve code structure and design

- more maintainable
- easier to understand
- easier to modify
- easier to add new features

Refactor

When?

Refactor when

- you add new features and the code is brittle or hard to understand.

  Refactoring makes this feature and future features easier to build.

- you fix bugs.

- during code review.

Refactor

"Smell check" your code!

- Duplication
- Unnecessary complexity
- Useless or misleading comments
- Long classes
- Long methods
- Poor names for variables, methods, classes
- Code that’s not used

Points to Ponder

- Cohesion
- Don’t Repeat Yourself (DRY)
- Tell Don’t Ask (TDA)

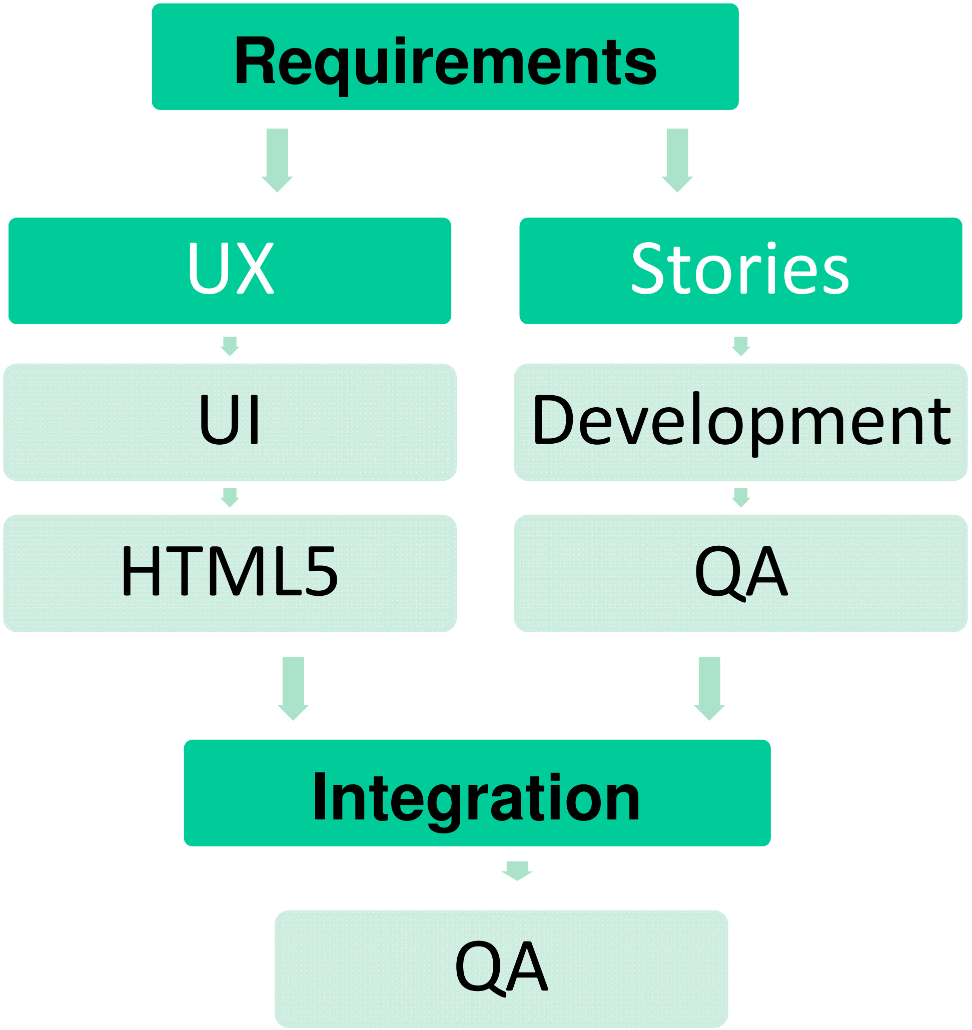
Dont do this:

```
function foo() {
   var i;
   // ...
   i = 5;
}
```

Do this:

```
function foo() {
   var i = 5;
   // ...
}
```

Protocol


Requirements
UX
Stories
UI
Development
HTML5
QA
Integration
QA

Thank You