Software Development

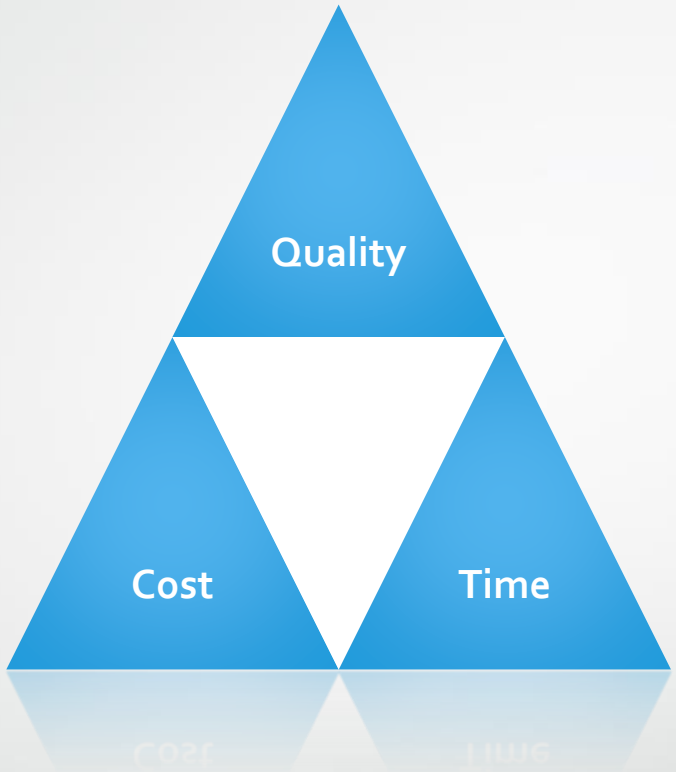
By Marcos Prieto, Software Engineer

Agenda

1. Foreknowledge
   - Reality Triangle
   - Buy vs Build
   - SLDC
   - Waterfall
   - Agile
2. Development
   - Open Source
   - HTML
   - CSS
   - JS
   - GIT

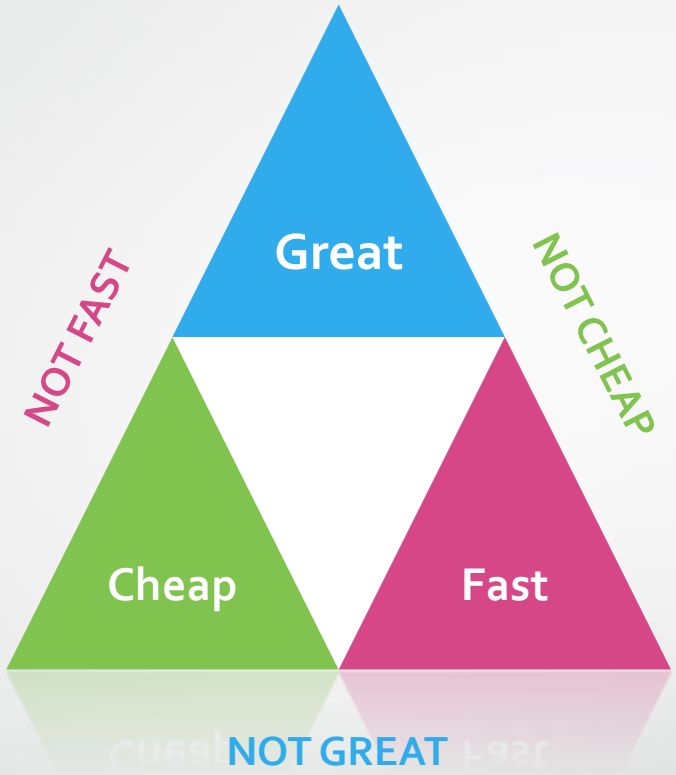
FOREKNOWLEDGE

what you need to learn before you learn



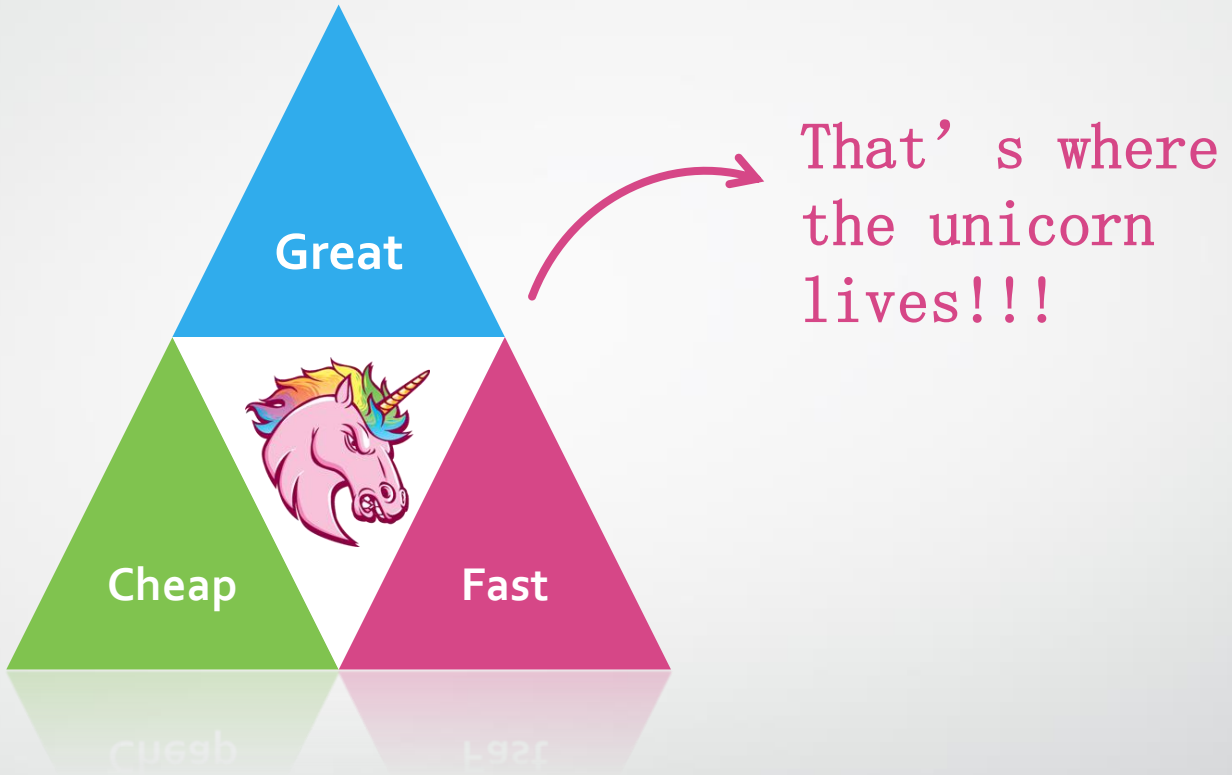
The Reality Triangle

three real-world constraints



The Reality Triangle

you can only pick two



The Reality Triangle

what if you pick three?

Buy a Mac

Pros

- Reliable: It just works.
- Intuitive: Customer first approach.
- Quality: Amazing! Fantastic! Insanely great!
- Improves your coolness factor.

Cons

- Extremely expensive!
- Closed and controlled.
- Difficult and costly to upgrade.
- Apple Kool-Aid.

Build a PC

Pros

- Complete control.
- Better hardware.
- Better price.
- Your choice of software.

Cons

- Time consuming and extremely technical.
- Higher potential for problems: Multiple components from various vendors being assembled by you.
- Limited customer support.

Buy a Mac

Pros

- Reliable: It just works.
- Intuitive: Customer first approach.
- Quality: Amazing! Fantastic! Insanely great!
- Improves your coolness factor.

Cons

- Extremely expensive!
- Closed and controlled.
- Difficult and costly to upgrade.
- Kool-Aid.

Build a PC

Pros

- Complete control.
- Better hardware.
- Better price.
- Your choice of software.

Cons

- Time consuming and extremely technical.
- Higher potential for problems: Multiple components from various vendors being assembled by you.
- Limited customer support.

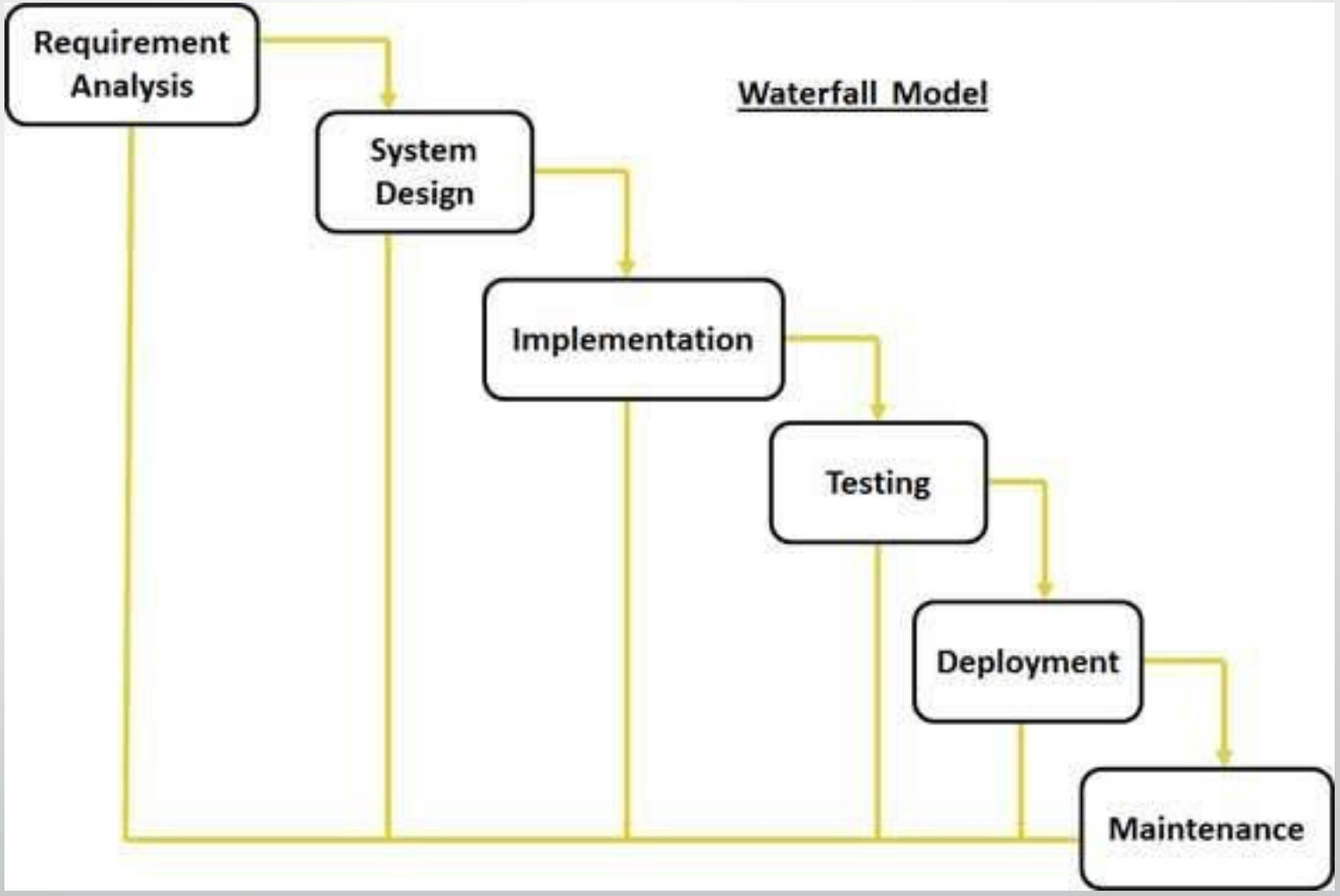
SDLC

software development life cycle

"The systems development life cycle (**SDLC**) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application."

google: define SDLC

Waterfall


Waterfall Model
Requirement Analysis
System Design
Implementation
Testing
Deployment
Maintenance

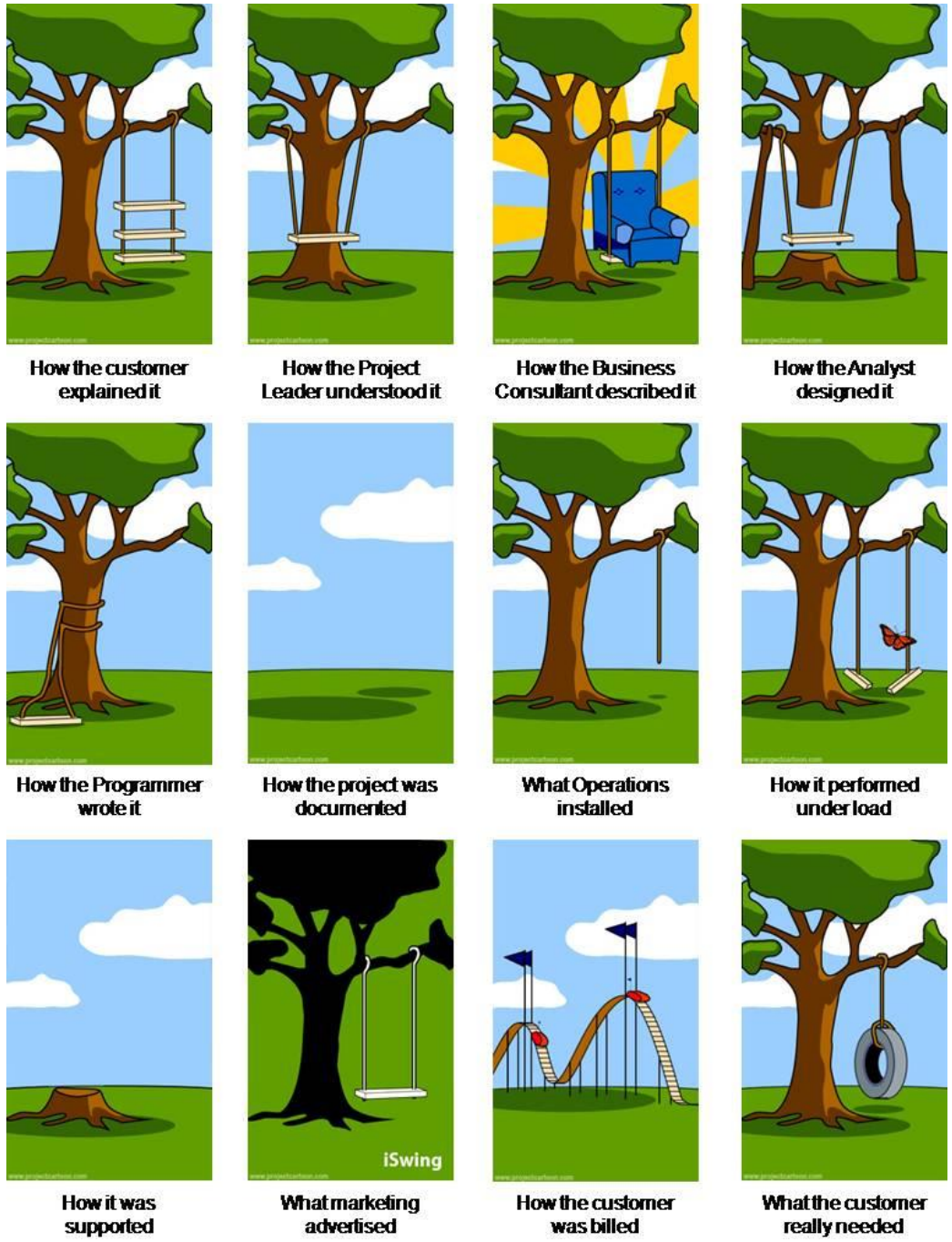
"Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially."

http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

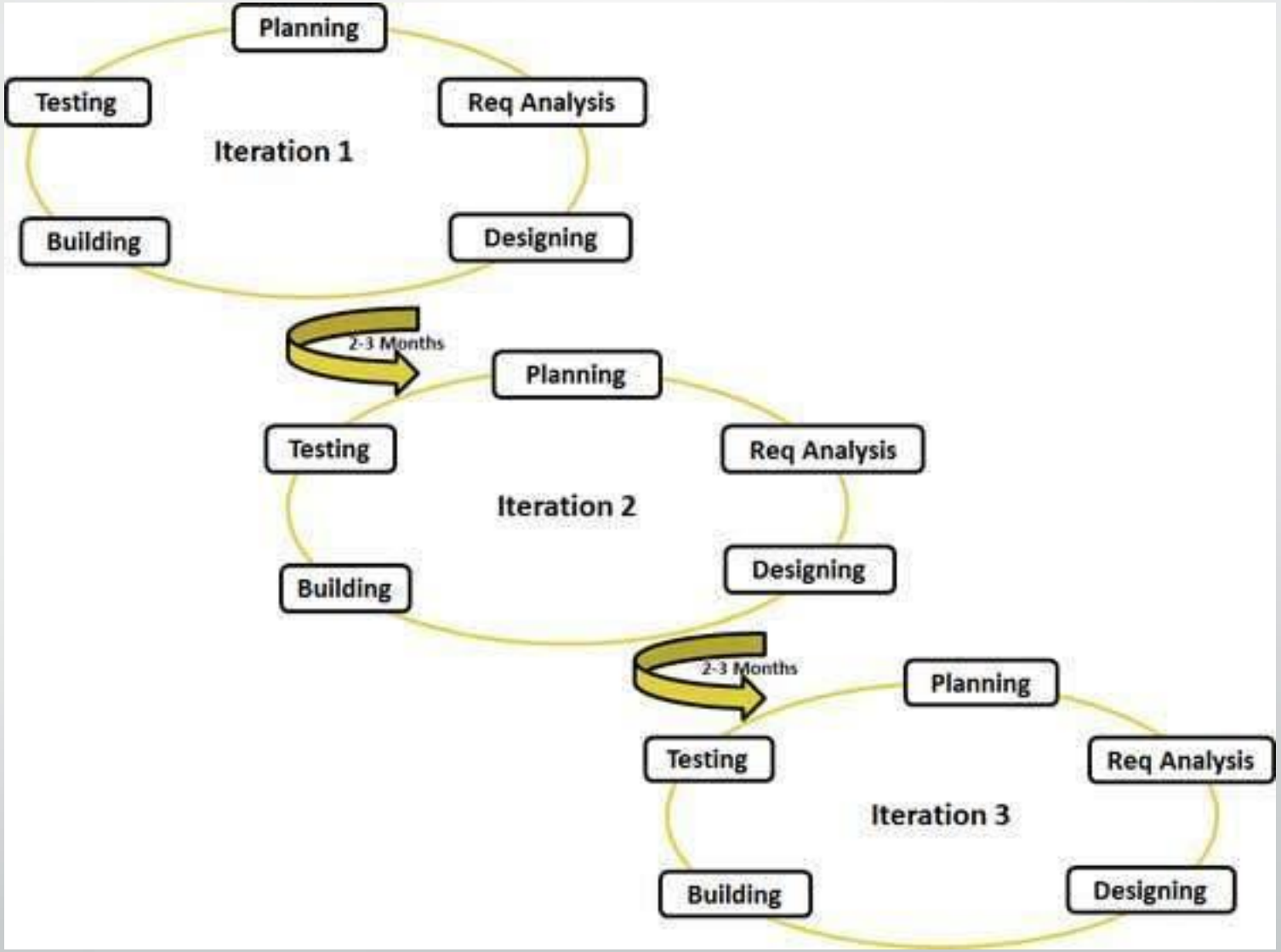
Waterfall

Waterfall

Linear; Silos



Agile


Planning
Testing
Req Analysis
Iteration 1
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 2
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 3
Building
Designing

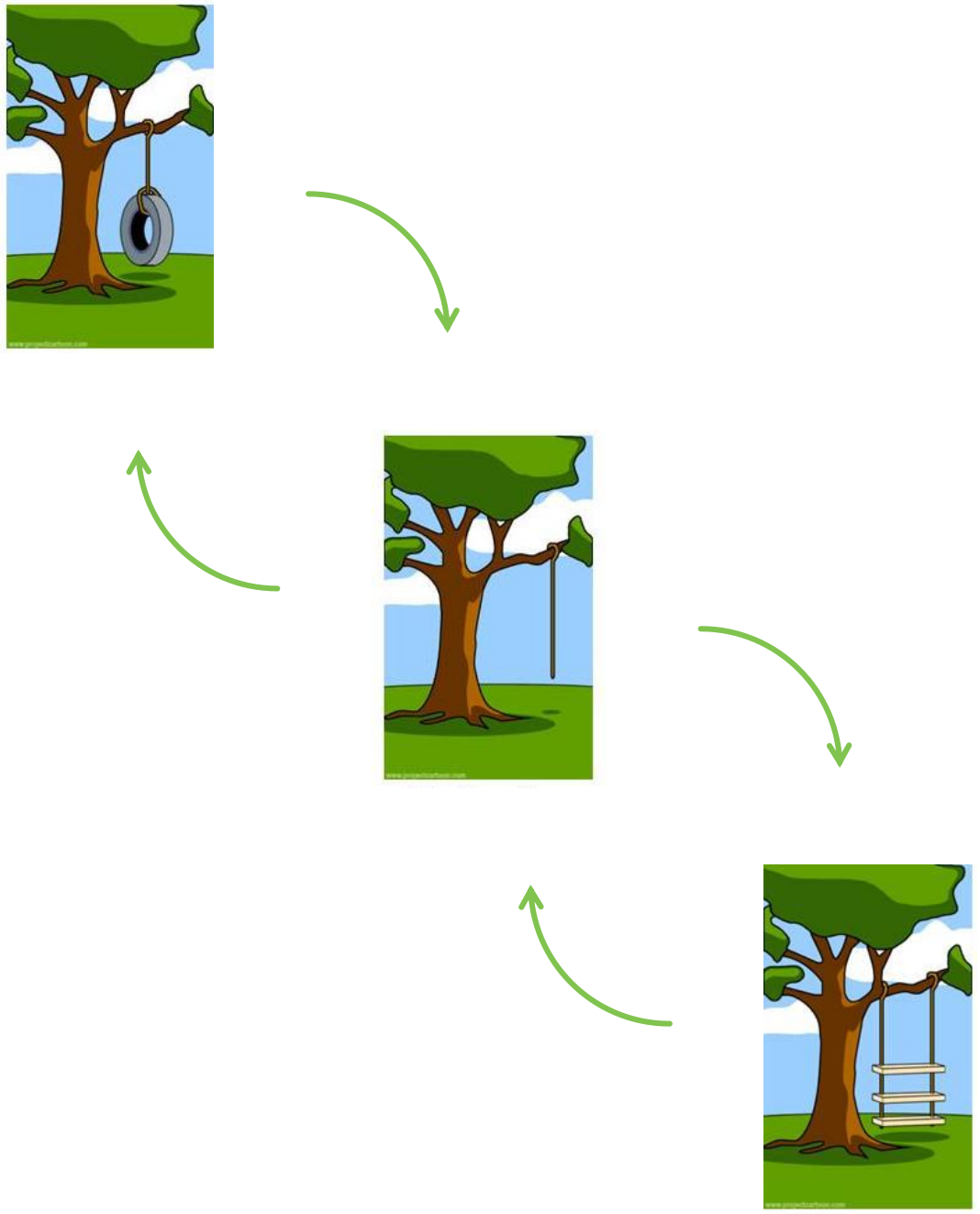
"Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer."

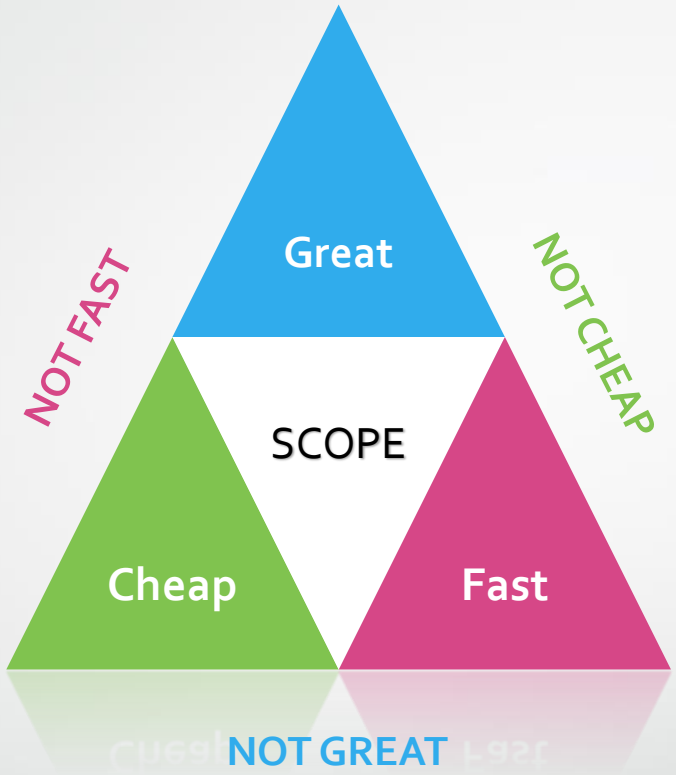
http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

Agile

Agile

Iterative; feedback loop





The Reality Triangle

with agile you can have all three because if you control the scope

External Product / Service

Pros

- Setup tends to be quicker, easier.
- Reliability: QA / SLA / no SPOF.
- Scalability: can grow / shrink with ease
- Dedicated internal resources not required

Cons

- Can be extremely expensive!
- Can be closed and controlled.
- Can be difficult and costly to upgrade.
- Can require annuity.
- Can be costly / slow to migrate to / from.

Internal Software Development

Pros

- Complete control.
- Develop exactly what you want / need.
- Less expensive (open source / no annuity).
- Easy to modify / update / upgrade.

Cons

- Time consuming and extremely technical.
- Bug/Error prone: requires lots of testing.
- Ties-up internal resources.
- Requires maintenance.
- Can fail.

External Product / Service

Pros

- Setup tends to be quicker, easier.
- Reliability: QA / SLA / no SPOF.
- Scalability: can grow / shrink with ease
- Dedicated internal resources not required

Cons

- Can be extremely expensive!
- Can be closed and controlled.
- Can be difficult and costly to upgrade.
- Can require annuity.
- Can be costly / slow to migrate to / from.

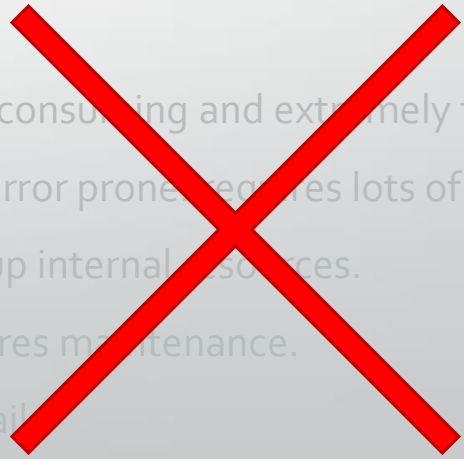
Internal Software Development

Pros

- Complete control.
- Develop exactly what you want / need.
- Less expensive (no annuity).
- Easy to modify / update / upgrade.

Cons

- Time consuming and extremely technical.
- Bug/Error prone requires lots of testing.
- Ties-up internal resources.
- Requires maintenance.
- Can fail.




Q
A

DEVELOPMENT

understanding the buzz words…

node.js™

Jasmine

git

Atlassian
SourceTree

HTML

CSS

JAVASCRIPT

Protractor

end to end testing for AngularJS


Se


node
JS

Jasmine


ANGULARJS


git


Atlassian
SourceTree


HTML
CSS
JAVASCRIPT


Se


Protractor
end to end testing for AngularJS

Browser – UI

DOM – the structure of a page

FILE → BROWSER → DOM

Browser – UI

HTML – the web's semantic building block

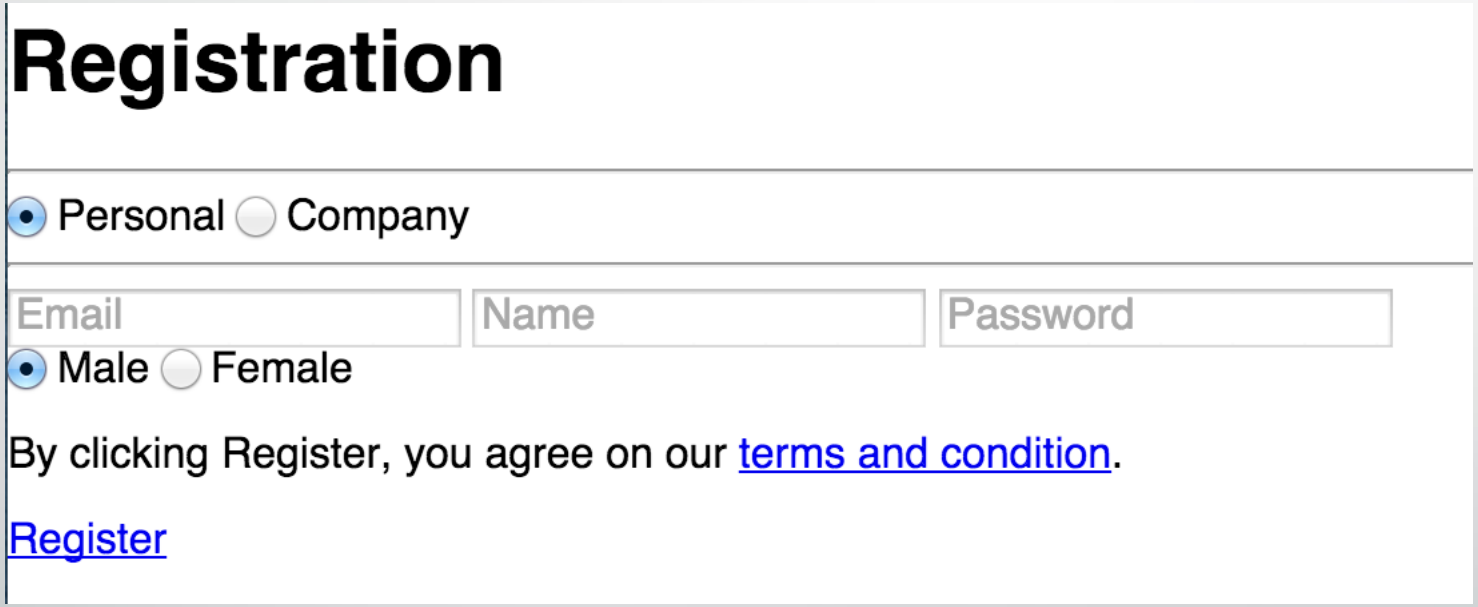
<tagname attribute="value"></tagname>

1. Go to Google
2. FYI…
3. <h1 id="FAQs">Frequently Asked Questions</h1>

http://www.w3schools.com/tags

Browser – UI

HTML – the web's semantic building block



Browser – UI

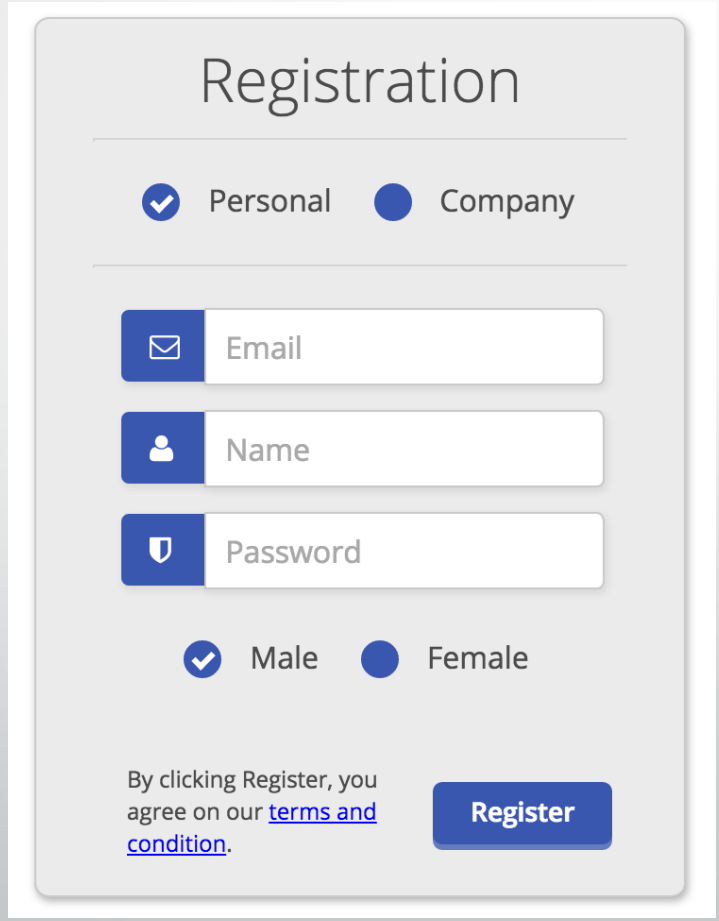
CSS – the web's interior design language

```
Selector {
   attribute name: value;
}
```

Target/Specificity: HTML tags > .classes > #IDs

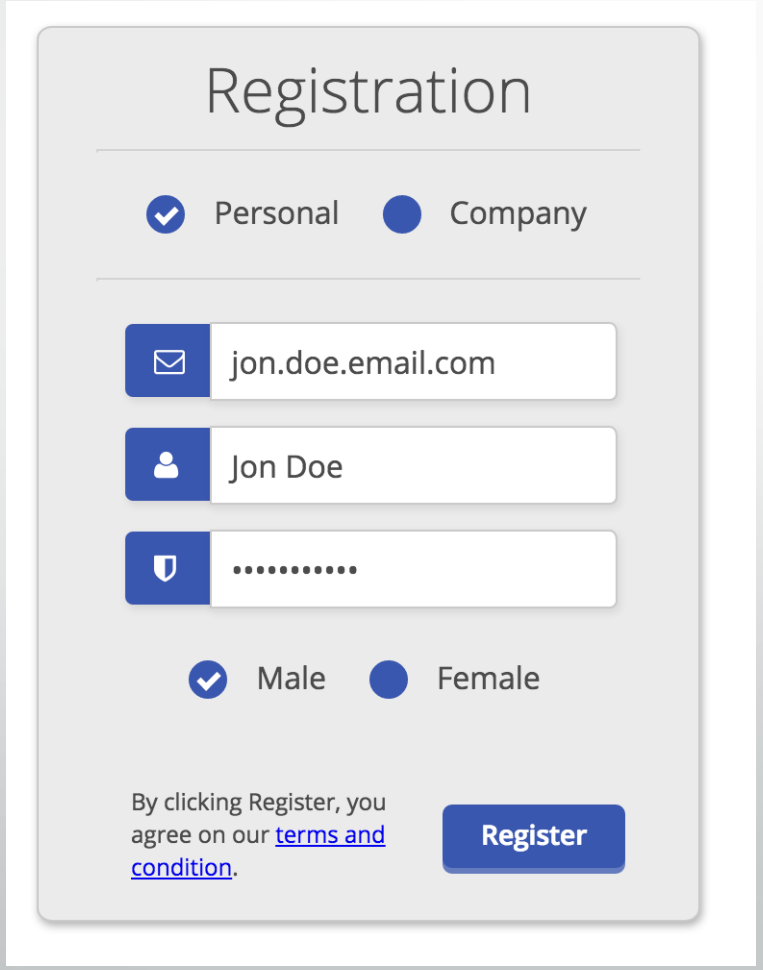
Browser – UI

CSS – the web's interior design language



Browser – UI

JS – the scripting language of the web



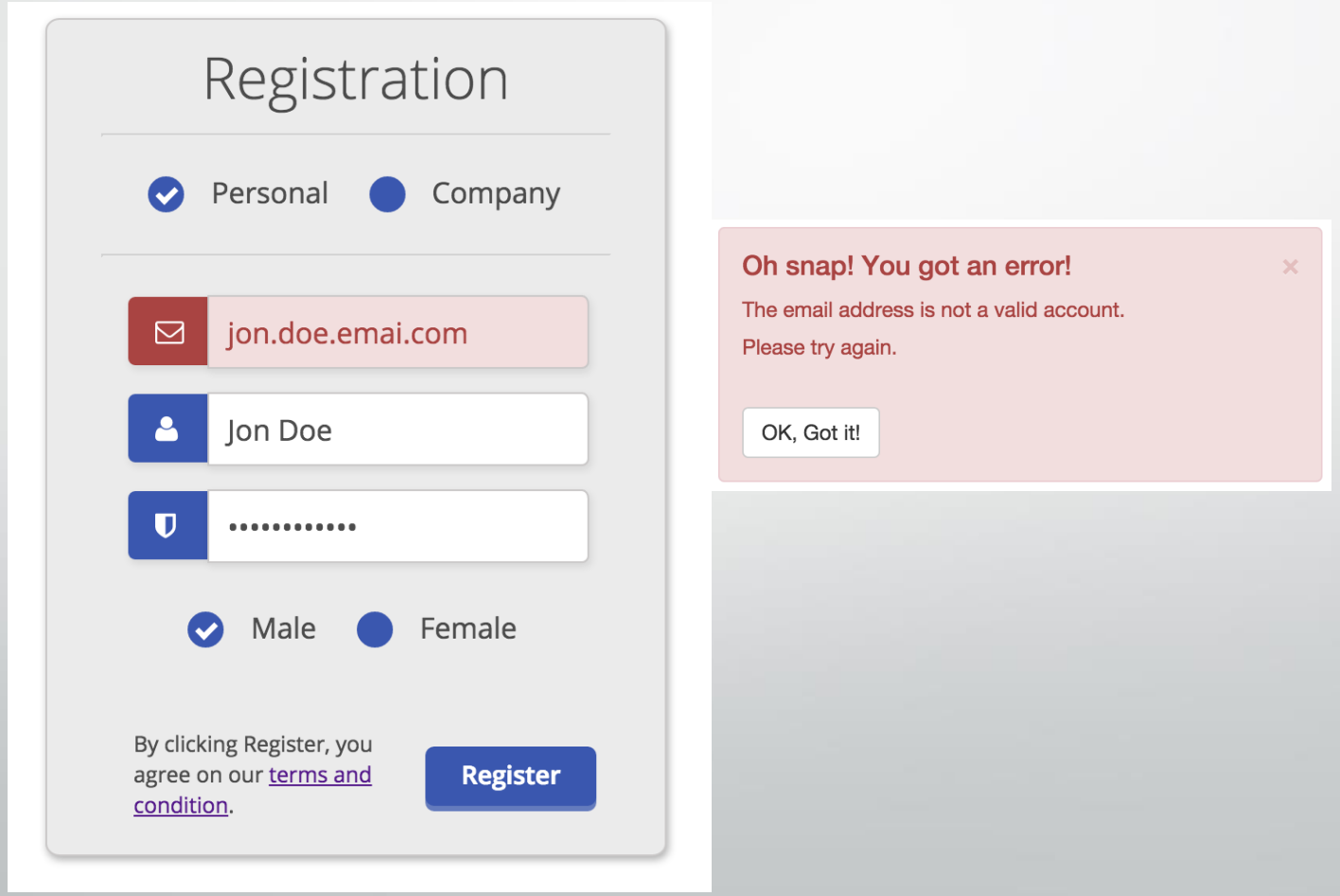
Browser – UI

JS – the scripting language of the web

```
1. function onFormSubmit() {
2.   var email = document.getElementById('email');
3.   if (isEmailAddressValid(email.value) === true) {
4.       displayRegisteredMessage();
5.   } else {
6.       displayErrorMessage();
7.   }
8. }
```

Browser – UI

JS – the scripting language of the web




Se


node
JS


Jasmine


ANGULARJS


git

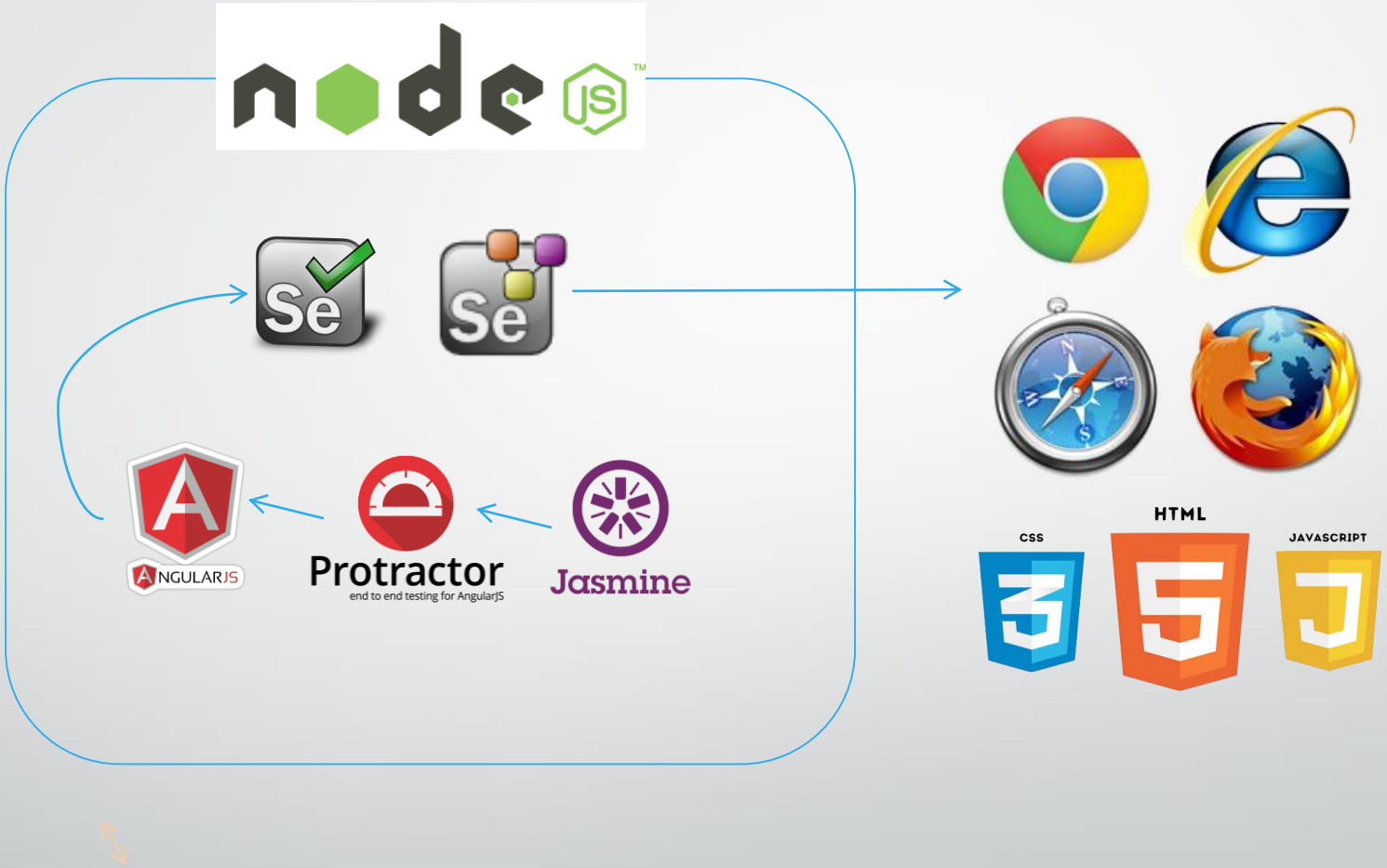

Atlassian
SourceTree


HTML
CSS
JAVASCRIPT


Se


Protractor
end to end testing for AngularJS

Tech Stack


node
JS
Se
Se
ANGULARJS
Protractor
end to end testing for AngularJS
Jasmine
CSS
HTML
JAVASCRIPT


Jasmine
ANGULARJS
git
Atlassian
SourceTree
CSS
HTML
JAVASCRIPT
Protractor
end to end testing for AngularJS

GIT

Code Repository – agile code development

git --distributed-even-if-your-workflow-isnt

Search entire site...

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Learn Git in your browser for free with Try Git.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads

Community

Latest source Release
2.3.4
Release Notes (2015-03-23)
Downloads for Mac


GIT BASICS
LEVEL 1

GIT

Try Git

Replay Course

COURSE DESCRIPTION

Learn how to use Git by reviewing the basic concepts of Git version control. Try out this introductory course that was created with GitHub.

COURSE OVERVIEW

ABOUT THE PROFESSORS

Gregg Pollack

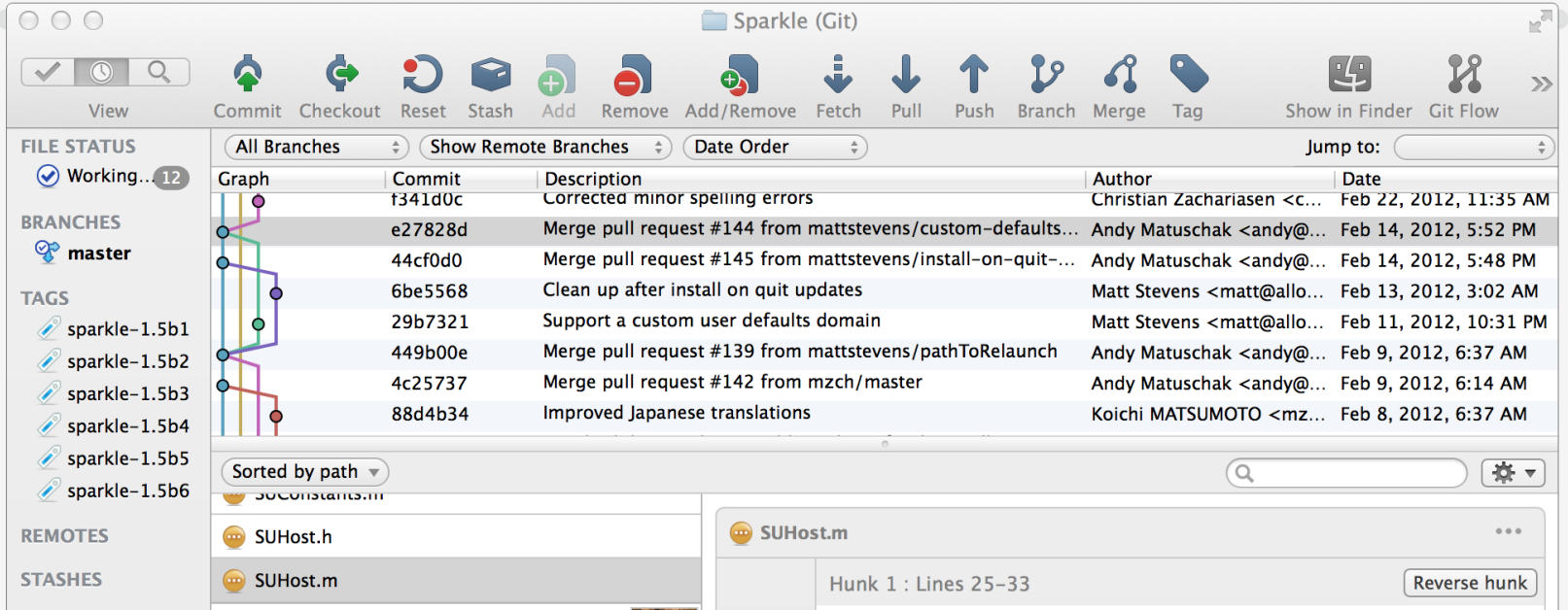
Gregg is passionate about taking complex topics and teaching them efficiently. He's helped build Envy Labs, Starter Studio, and Code School. He

SourceTree

A Graphical User Interface for GIT

Atlassian SourceTree

A free Git & Mercurial client for Windows or Mac.



Tech Stack


node
Se
Se
ANGULARJS
Protractor
end to end testing for AngularJS
Jasmine
CSS
HTML
JAVASCRIPT
git
Atlassian
SourceTree


Q
A

Accounting Architecture

…how this relates to you.

Extra-credit presentations

1. Thoroughly describe what occurs in each step on the Software Development Life Cycle. Use examples, where possible.
2. Explain Agile and give examples of Agile methodologies.
3. Define DevOps and Continuous Improvement. Focus especially on how Accounting Architecture helps accountants to participate in DevOps.
4. Introduce the testing tools referenced in the presentation: Selenium, Protractor, Jasmine. Explain how node.js helps to facilitate these tools.

Software Development

By Marcos Prieto, Software Engineer