Career Story

Hector Correa
(he/him/his)

My career in four buckets

College and immediately after college (1991-1998)

- Went to college in Colima, Mexico
- Graduated in 1996
- Computer Science

- Worked part time developer during college, full time after that

- Technology stack
  - At school: Assembler, Pascal, C/C++, COBOL, VAX, IBM/360, Db2
  - At work: Clipper, FoxBase, MS-DOS, Windows, Novell

My career in four buckets

College and immediately after college (1991-1998)

- Went to college in Colima, Mexico
- Graduated in 1996
- Computer Science

- Worked part time developer during college, full time after that

- Technology stack
  - At school: Assembler, Pascal, C/C++, COBOL, VAX, IBM/360, Db2
  - At work: MS-DOS, Windows, Novell, Clipper, FoxBase

***Client server apps (*1998-2005)**

- Moved to the US in 1998
- Bunch of consulting work (insurance, engineering, banking)
- Worked on *client-server* applications for Windows
- Technology stack
  - Windows, Microsoft FoxPro, and Microsoft SQL Server
  - Got certifications on Microsoft stuff

My career in four buckets

College and immediately after college (1991-1998)

- Technology stack
  - At school: Assembler, Pascal, C/C++, COBOL, VAX, IBM/360, Db2
  - At work: MS-DOS, Windows, Novell, Clipper, FoxBase

Client server apps (1998-2005)

- Technology stack
  - Windows, Microsoft FoxPro, and Microsoft SQL Server

Web applications (2005-2014)

- Client-server and *web applications* for
  - bankruptcy management (Chapter 11)
  - electronic discovery
  - tax preparation
  - weather forecasting
- Technology stack
  - Windows, Microsoft C#, Microsoft SQL Server, Oracle, ASP.NET
  - Certifications/training on Agile software development

My career in four buckets

College and immediately after college (1991-1998)

- Technology stack
  - At school: Assembler, Pascal, C/C++, COBOL, VAX, IBM/360, Db2
  - At work: MS-DOS, Windows, Novell, Clipper, FoxBase

***Client server apps (*1998-2005)**

- Technology stack
  - Windows, Microsoft FoxPro, and Microsoft SQL Server

***Web applications (*2005-2014)**

- Technology stack
  - Windows, Microsoft C#, Microsoft SQL Server, Oracle, ASP.NET

Library-land (2014-present)

- Took my first job at a library
- Penn State ➡ Brown ➡ Princeton
- Technology stack
  - Linux/Mac, Ruby on Rails, PostgreSQL, MySQL, Solr

My career in four buckets

College and immediately after college (1991-1998)

- Technology stack
  - At school: Assembler, Pascal, C/C++, COBOL, VAX, IBM/360, Db2
  - At work: MS-DOS, Windows, Novell, Clipper, FoxBase

***Client server apps (*1998-2005)**

- Technology stack
  - Windows, Microsoft FoxPro, and Microsoft SQL Server

***Web applications (*2005-2014)**

- Technology stack
  - Windows, Microsoft C#, Microsoft SQL Server, Oracle, ASP.NET

Library-land (2014-present)

- Took my first job at a library
- Penn State ➡ Brown ➡ Princeton
- Technology stack
  - Linux/Mac, Ruby on Rails, PostgreSQL, MySQL, Solr

Automatic Code Generation
4GL/5GL
UML
Rational Rose
n-tier architecture
Palm Pilot
Object Oriented Databases
Entity Framework
Visual Basic
ActiveX
SOAP
XAML
Visual Source Safe
LINQ
MongoDB
Express.js
Angular
OLE/COM
WCF
CoffeeScript
ADO
WPF
LDP
Fedora Commons
IIS

A few observations…

1. Learning

- There is always something new to learn in this field
- Learn things as they are meant to be used
  - Do: Learn Java as Java, Ruby as Ruby
  - Don’t: Learn Java as Ruby (or vice-versa)
- Write and present at conferences as a way to learn
  - Forces you to learn
  - Gives you exposure
- It is *never* too late to learn a new tool, technology, theory

2. Tools and theory

- The *tools* that I learned in school…
  - …are mostly obsolete by now

- But *theory* and principles still apply
  - Data structures, databases, software design, operating systems, graph theory, artificial intelligence

- We, as developers, tend to focus a lot on tools
  - It pays to know the tools (latest version of Ruby, Java, PostgreSQL)
  - Employers/peers look for this
  - *Become very good at them*
    - *…and be willing to ditch them*

3. Programming and problem solving

- Programming is about problem solving
- Software development is about problem solving
  - …with others
  - …with unclear and changing requirements
  - …that will require maintenance
- Technology is political
  - Data, algorithms, and the systems we write expose and amplify biases

4. You’ll screw something big at some point

- e.g. Delete the wrong data, take a system down accidentally
- Happens to all of us, more than once
- Don’t let it define you, use it as a learning opportunity

5. Programming is fun and rewarding

- You are always learning
- Opportunity to work in many domains
  - Cultural, engineering, finance, legal, …
- Make a difference in people
- It pays well

Good luck in your journey!