

# Software Development

in Constant Change

# Development Stack

- IDE: IntelliJ
- Languages: Java, Javascript
- Frameworks: Spring and ReactJS
- Agile: Scrum and Kanban
- Test Driven Development/Test Coverage
- Project Management: JIRA

# Frameworks

- Spring
  - Useful for quickly standing up a webserver
  - Dependency injection is top notch
  - Easy debugging using IntelliJ
  - Connect components
  - Label them
  - Control their life cycles/caching
  - Alter behavior based on configuration
- ReactJS
  - Build user interfaces specifically for single page applications
  - React allows developers to create large web applications which can change data, **without reloading the page**
  - Properties flow down; actions flow up
  - JSX is simple JavaScript which allows HTML quoting and uses these HTML tag syntax to render subcomponents

# Agile Methodologies

- Scrum
  - Product Owner
  - Scrum Master
  - Product Backlog
  - Work in sprints (normally 2 weeks)
  - Velocity is the key metric
  - Daily scrum - What you did yesterday, what you plan to do today, and any obstacles that you face
  - Sprint Review: Product Inspection
  - Sprint Retrospective: Process Inspection

# Agile Methodologies

- Kanban
  - Work is always visible
  - Daily standup - walk the board - “How can we get things to Done?”
  - Cycle-time is the key metric
  - Work in Progress (WIP) limit multi-tasking
  - Often associated with continuous delivery/continuous integration

# Agile Methodologies

	SCRUM	KANBAN
Cadence	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint if approved by the product owner	Continuous delivery or at the team's discretion
Roles	Product owner, scrum master, development team	No existing roles. Some teams enlist the help of an Agile coach.
Key metrics	Velocity	Cycle time
Change philosophy	Between sprints	Can happen at any time

# Test Driven Development

1. Write requirements
2. Write tests for the requirements
3. Write the software to pass the tests

# Test Driven Development

- Don't wait to write the tests; they won't get written!
- Ensures that you have complete test coverage all of the time
- Come to agreement with the product owner on what should happen
- Gives reassurance when it is time to refactor



# Interviews

- Code reviews
- Coding skills
- Applied algorithm knowledge
- Architecture
- Mock interviews
- Soft skills

# Managing Your Career

- Be useful
- Show up
- Find mentors
- Work with good people
- Be good to work with!
- Be careful in your commitments
- Github repo
- LinkedIn account
- Contribute to open source
- Have a stack overflow account and answer questions!

# Software developer at an existing company

- SW Version 1.0: Does the job just barely, but it makes money!
- Your job: keep version 1 running, build version 2
- Common failure pattern: Rewrite the software completely!
- Common OK pattern: let me break off this small chunk into a microservice
- Common success pattern: I'll put an API over this chunk of functionality, then then rewrite the code within THEN make it a microservice

# Learn How to Learn

- Spend time reading code
- No, seriously: spend time reading code
- Read the classics: Mythical Man Month, Code Complete, anything by Dijkstra
- Read, ask questions and answer questions on stack overflow
- Go to conferences: first as an attendee, then as a speaker
- Practice doing hard things

# Getting a Job

- Do internships
- Make friends and keep in touch
- Find a company that interests you and meet someone who works there
- Prepare for the interview!
- Find out what their problems are and be prepared to talk about how you can help
- Have a resume that fits on one page
- Once you get the job, your past accomplishments are less important

# How to Think Like Management

Money -> features

Money -> development -> features

Make your boss's life easier

Measurable #s make management happy

Tying the measurable #s to money makes management happier

# Product, Program and Engineering

Product - What should we build?

Engineering - How should we build it?

Program - How should we deliver it?

# Career Progression

- BS in Electrical Engineering (1999)
- Developer -> Lead developer non-real time Sonar at Applied Research Labs (2000 - 2006)
- Developer -> Team Lead -> Manager Precision geolocation at Ticom Geomatics (2006 - 2013)
- MS in Software Engineering (2006)
- Manager -> Senior Manager -> Director APIs at Bazaarvoice (2013 - 2018)
- MBA (2014)
- Senior Manager Job Search Front End at Indeed (2018 - Present)



# Contact Info

[kurtis.white@gmail.com](mailto:kurtis.white@gmail.com)



Kurtis White



All

Images

Shopping

News

Videos

More

Settings

Tools

About 37,600,000 results (0.55 seconds)

**Kurtis White - Senior Manager - Indeed.com | LinkedIn**

<https://www.linkedin.com/in/kurtis-white-5a13aa39>

View **Kurtis White's** profile on LinkedIn, the world's largest professional community. Kurtis has 4 jobs listed on their profile. See the complete profile on LinkedIn ...