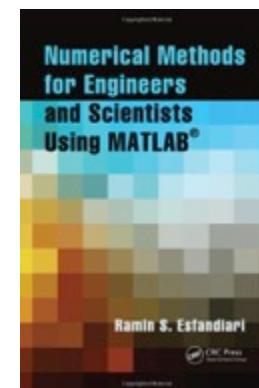
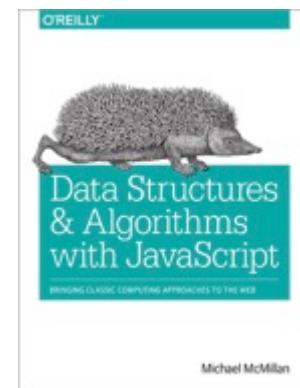
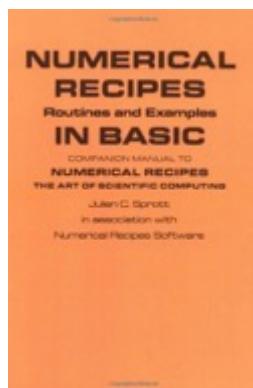
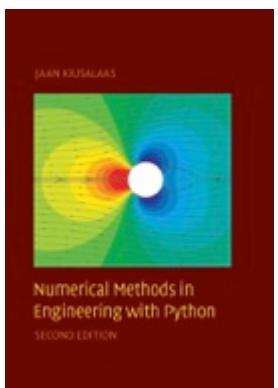
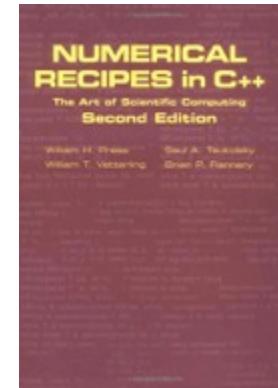
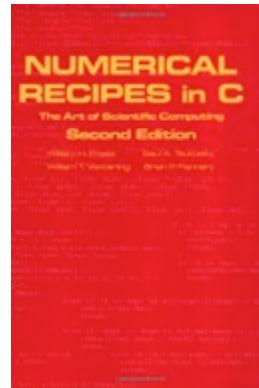
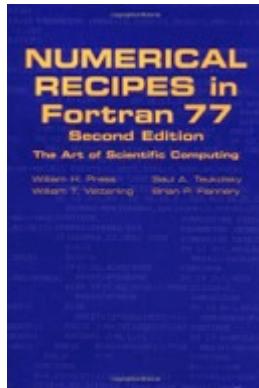
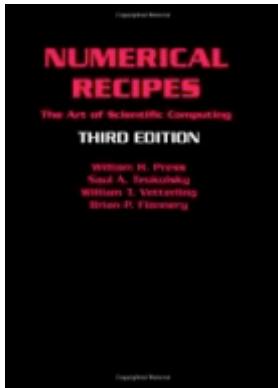


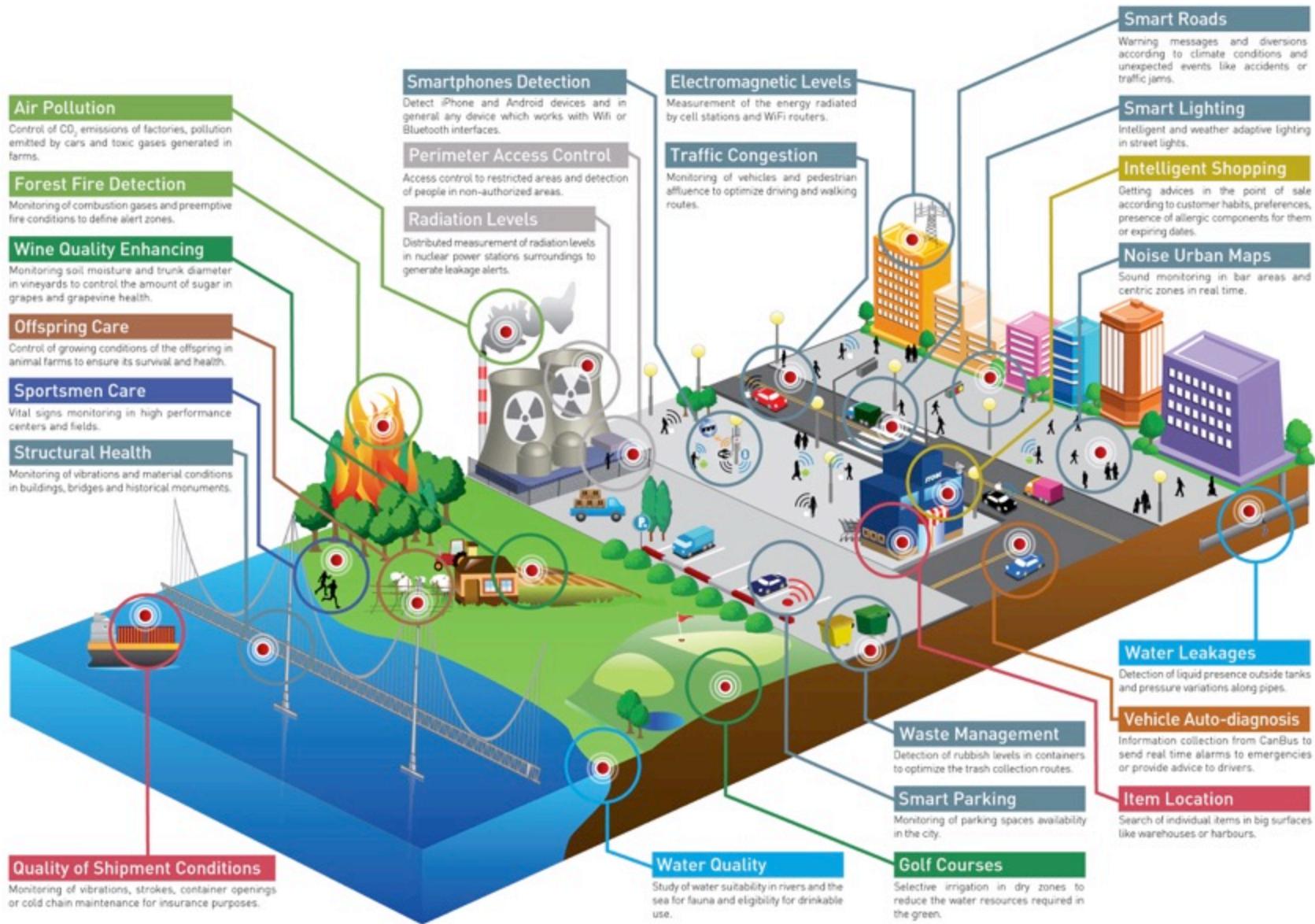
A white Westfalia van is shown from a rear three-quarter perspective, driving away from the viewer across a vast, flat landscape. The ground is covered in dry, golden-brown grass. In the background, there are several layers of mountains, also bathed in a warm, golden light. The sky above the mountains is a pale yellow, transitioning to a darker orange and then to a deep blue at the very top of the frame.

What Is
Different?

The Historical Approach



The Challenge: Complex Software Systems



The Challenge: Complex Software Systems



A white Westfalia van is shown from a rear three-quarter perspective, driving away from the viewer across a vast, flat landscape. The ground is a mix of dark earth and bright, glowing sand or dust. In the distance, a range of mountains is visible under a hazy, warm sky.

System Building Blocks

Application Servers:

- Small
- Easy to use
- Low Cost
- Ubiquitous
- Loosely Joined

Node.JS Ecosystem



Application Server Skills

- Service Creation
- Service Consumption
- Service Integration
- Data Transformations
- Concurrency
- Coordination



Package Management

- Registry
- Creation
- Distribution
- Scalability

Parallel Happens – Async Programs

- Non-Blocking
- Event-loop
- Callbacks
- Promises



State Management

- Record and Replay
- Predictable Containers
- Logging
- Time Travel



Event Streams

- Event Management
- Event bubbling
- Event Driven Programming
- Reactive Patterns

Coordination

- Orchestration
- Integration
- Error Handling



REST

- Microservices
- Route Mgmt.
- Token Mgmt.
- API Design



Containers

- Container Mgmt.
- Docker
- Mesos
- Kubernetes

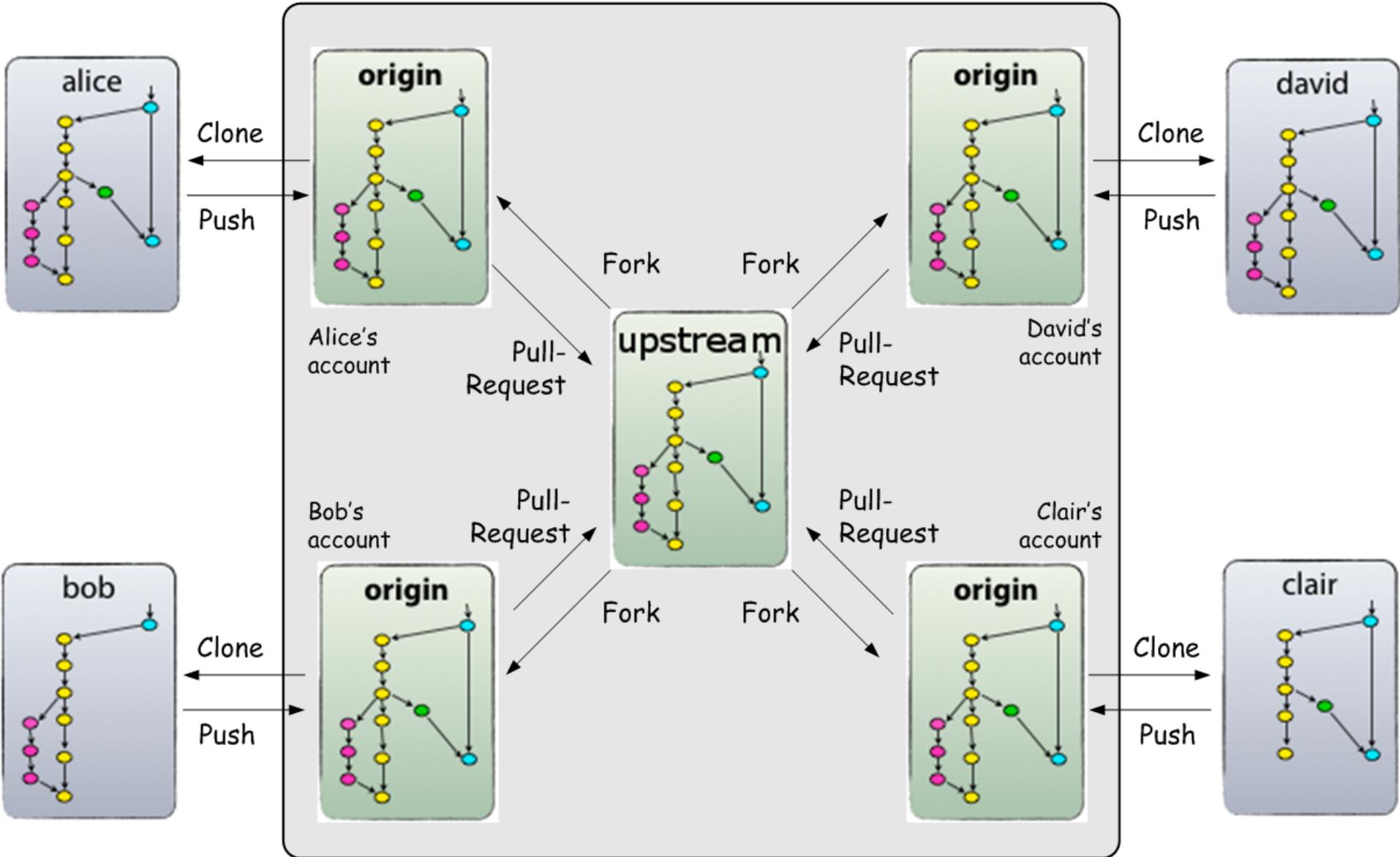


Collaboration

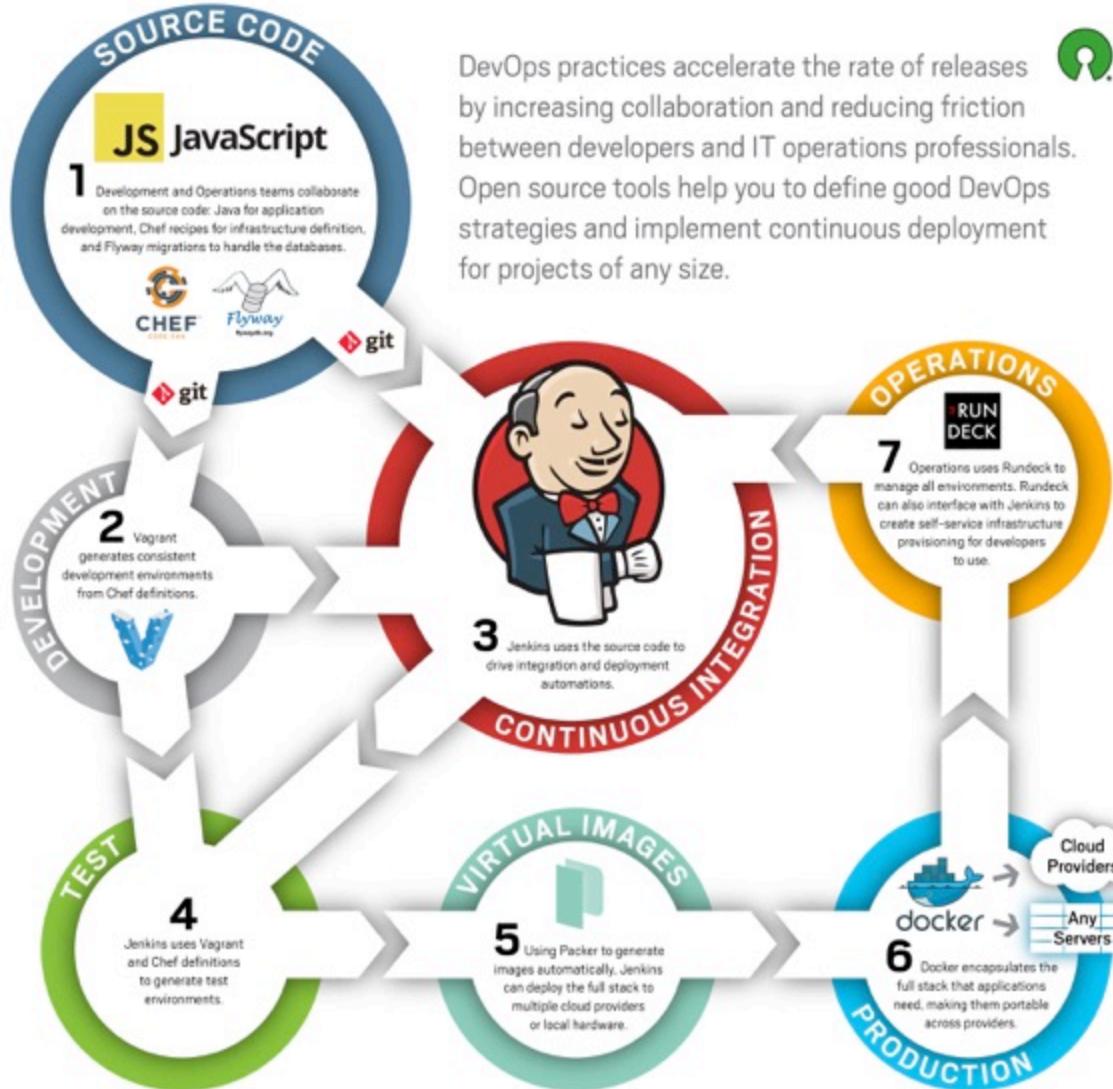
- Issues
- Tracking
- History
- Notifications
- Visualization
- Patterns



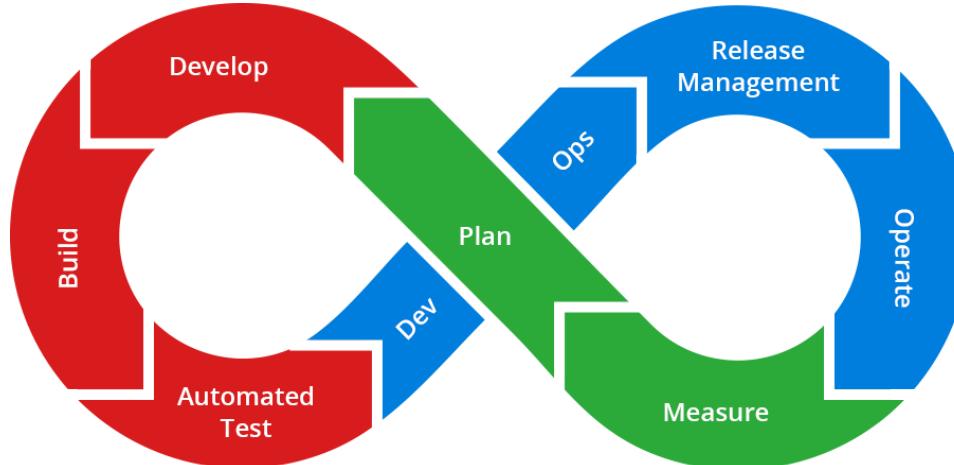
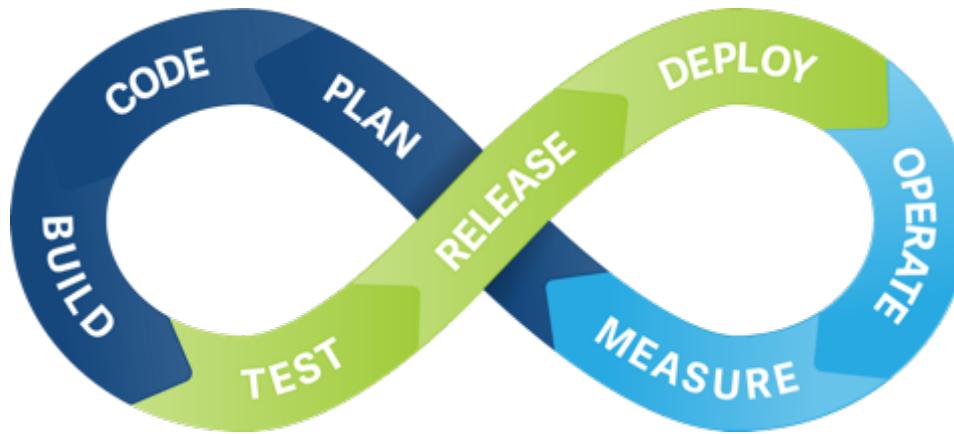
Collaboration



Continuous Integration



Continuous Integration



Continuous Integration

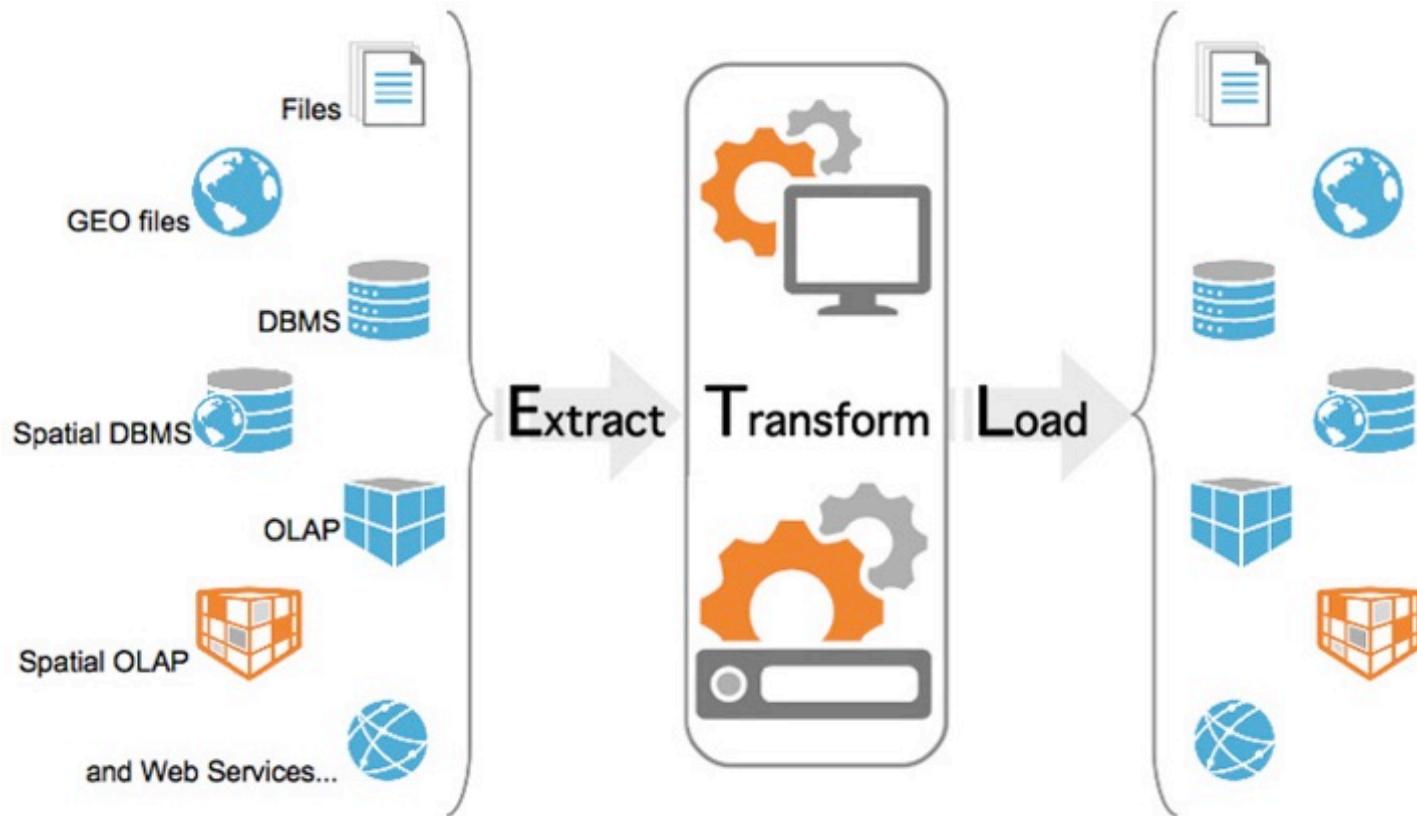
- Pull Requests
- Branch Select
- Notification
- Testing
- Containers
- Deployment



Big Data

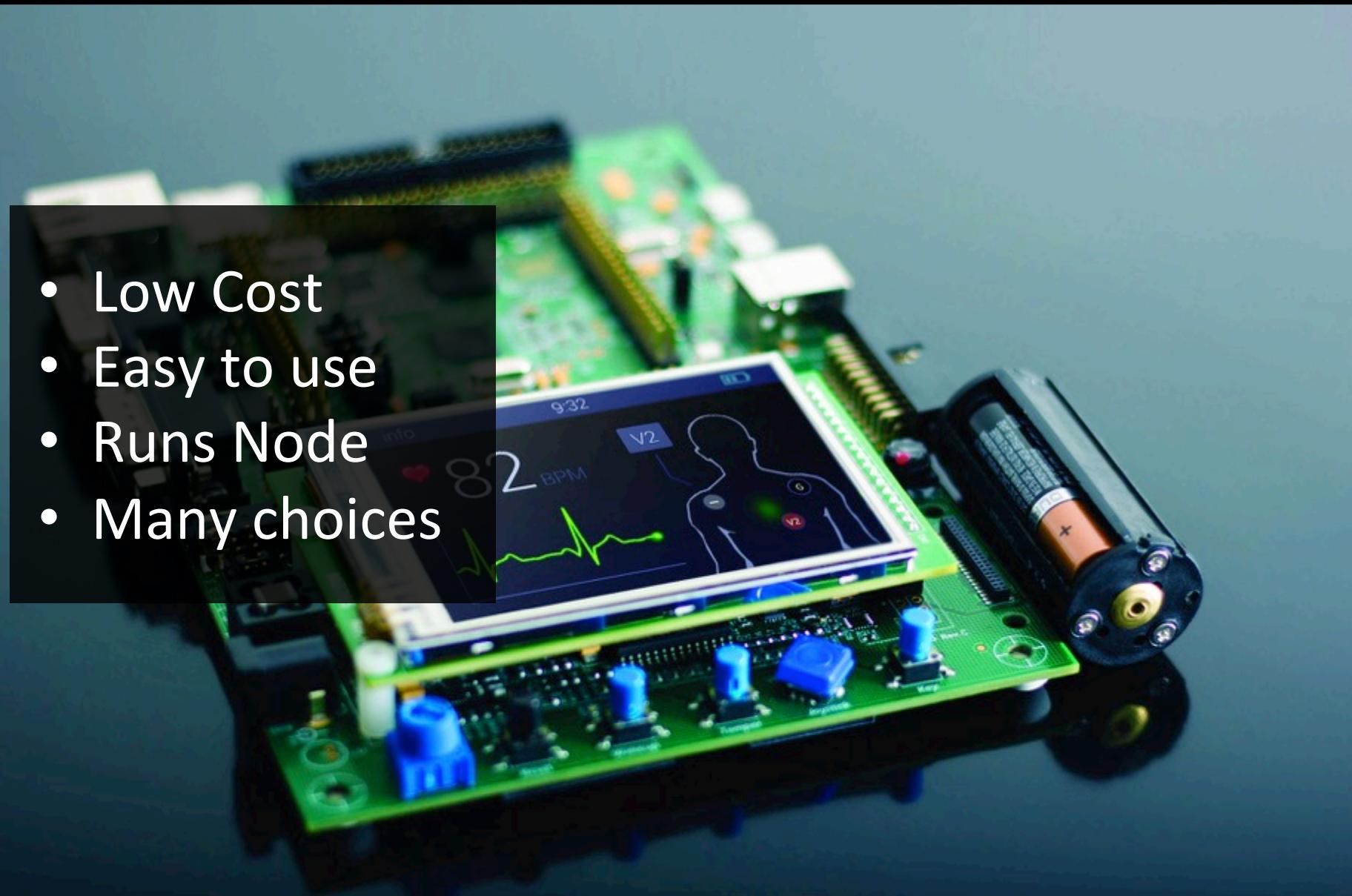
- Volume
- Variety
- Velocity
- NoSQL

Data Transformations



Commodity Hardware

- Low Cost
- Easy to use
- Runs Node.js
- Many choices



Security

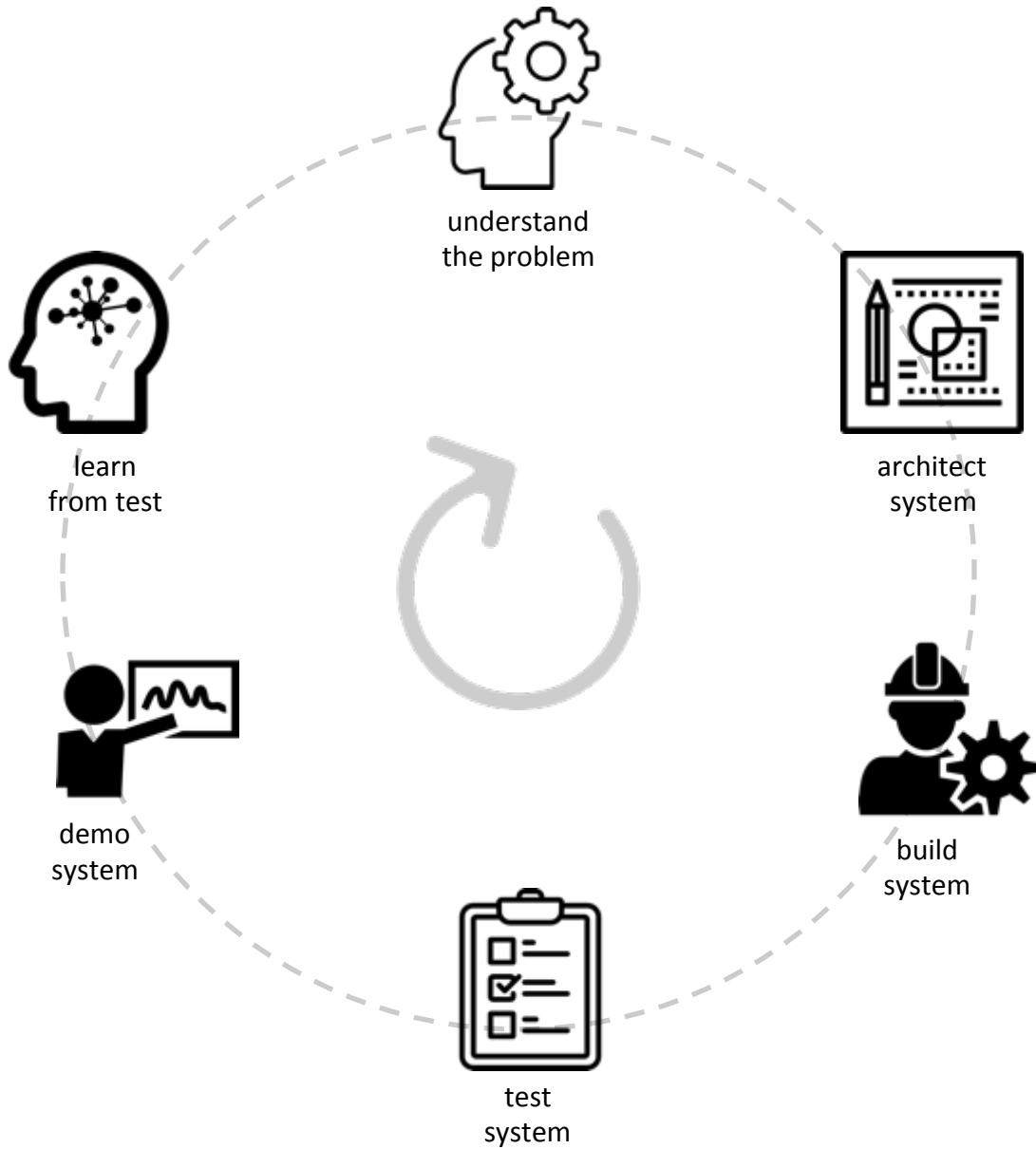
- Identify
- Protect
- Detect
- Respond
- Recover



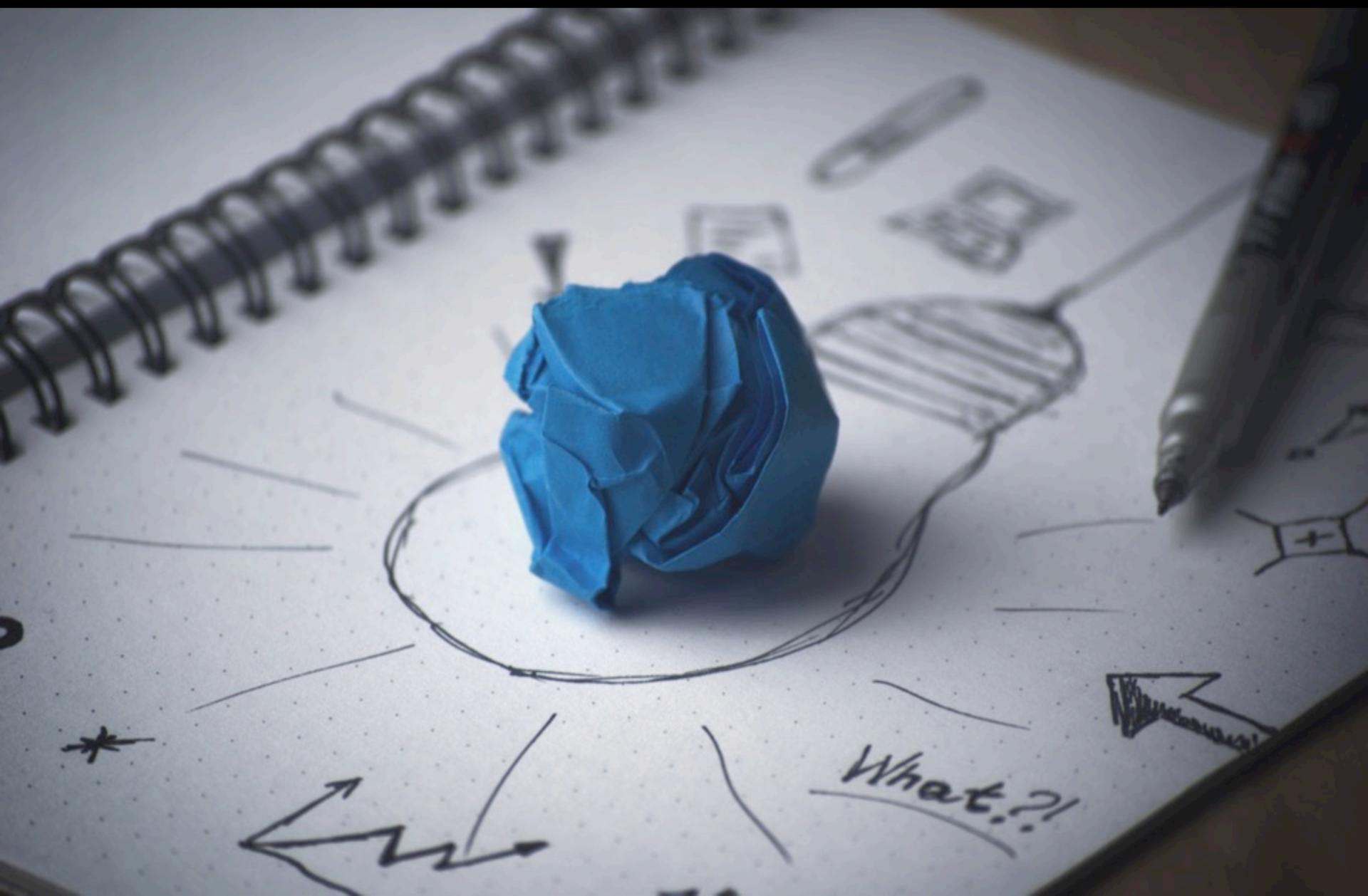
A white Westfalia van is shown from a rear three-quarter perspective, driving away from the viewer across a vast, flat, arid landscape. The van is kicking up a large cloud of dust behind it. The sky is a warm, golden-yellow color, suggesting either sunrise or sunset. In the distance, there are low, rolling hills or mountains. The overall atmosphere is one of travel and exploration.

Construction Cycle

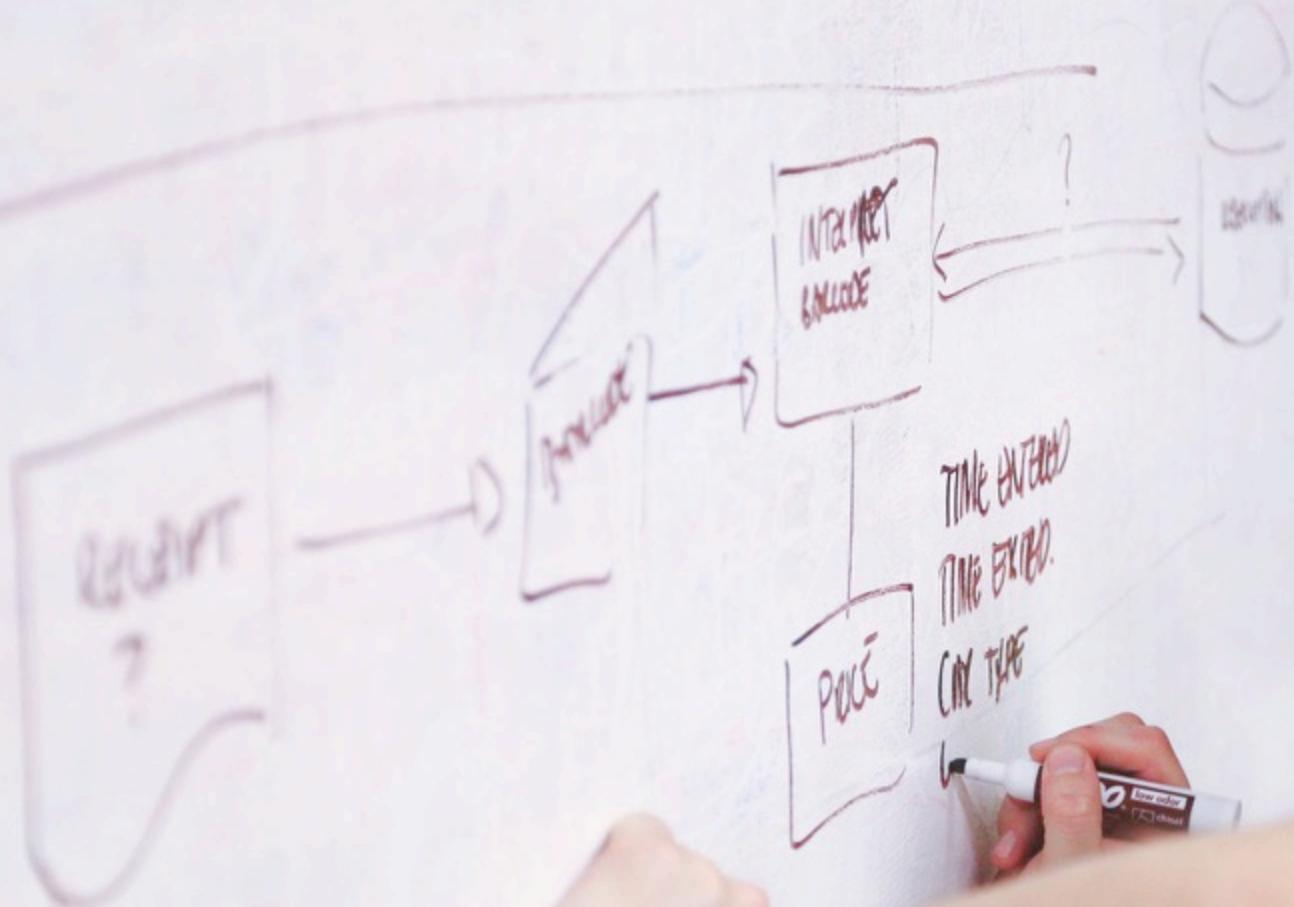
Complex Software System Cycle



Step 1: Understand the problem deeply

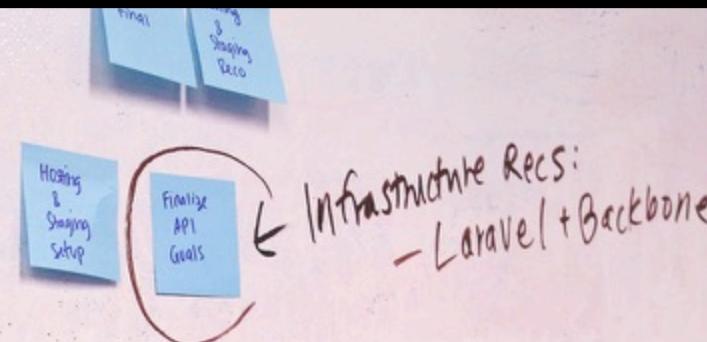


Step 2: Architect System

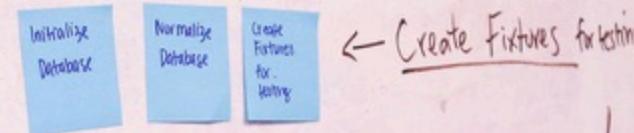


Step 3: Build System

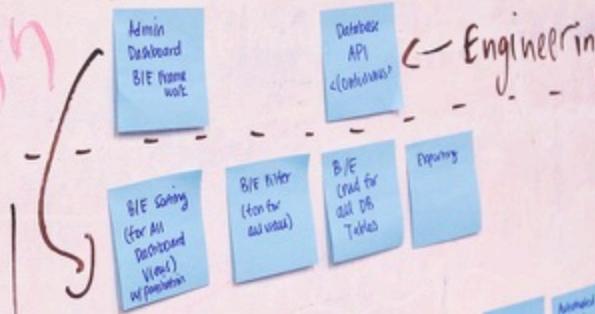
WEEK 3 : Design



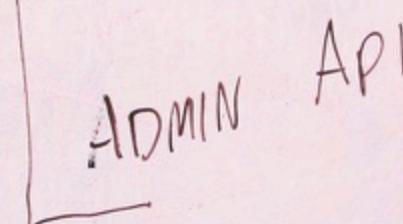
1) WEEK 4: Design



WEEK 5: Design



WEEK 6: Dev



WEEK 7: DEV

Step 4: Test the System



Step 5: Demo System



Step 6: Learn



Step 7: Iterate

