

Continuous Delivery in 90 Days

NARASIMHA BADRINATH

First Application

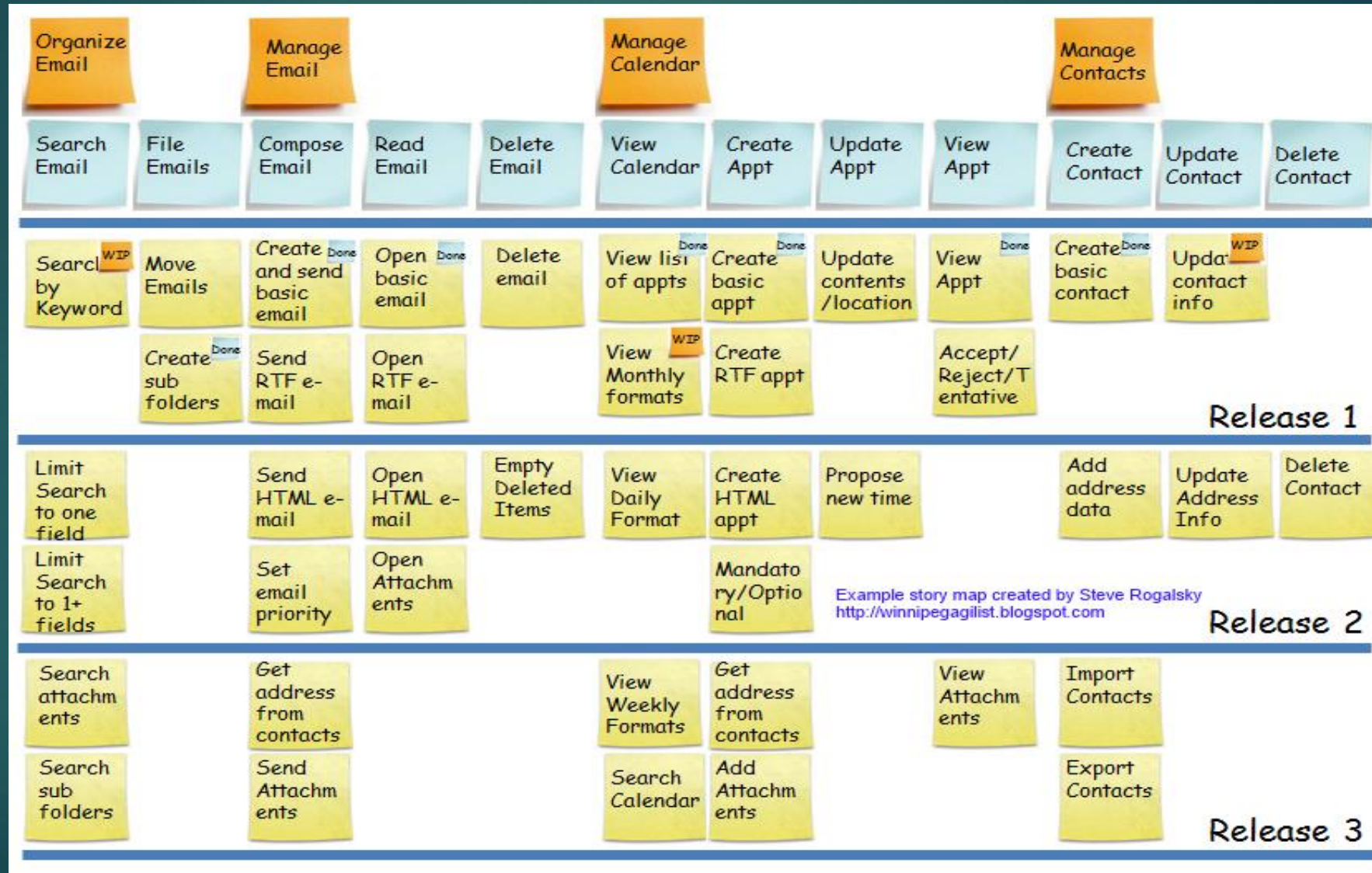
- ▶ Code Quality
- ▶ Product Quality
- ▶ QA, Acceptance, Production environments

What we tried..

Story mapping

A user **story map** arranges user stories into a useful model to help understand the functionality of the system, identify holes and omissions in your backlog, and effectively plan holistic releases that deliver value to users and business with each release

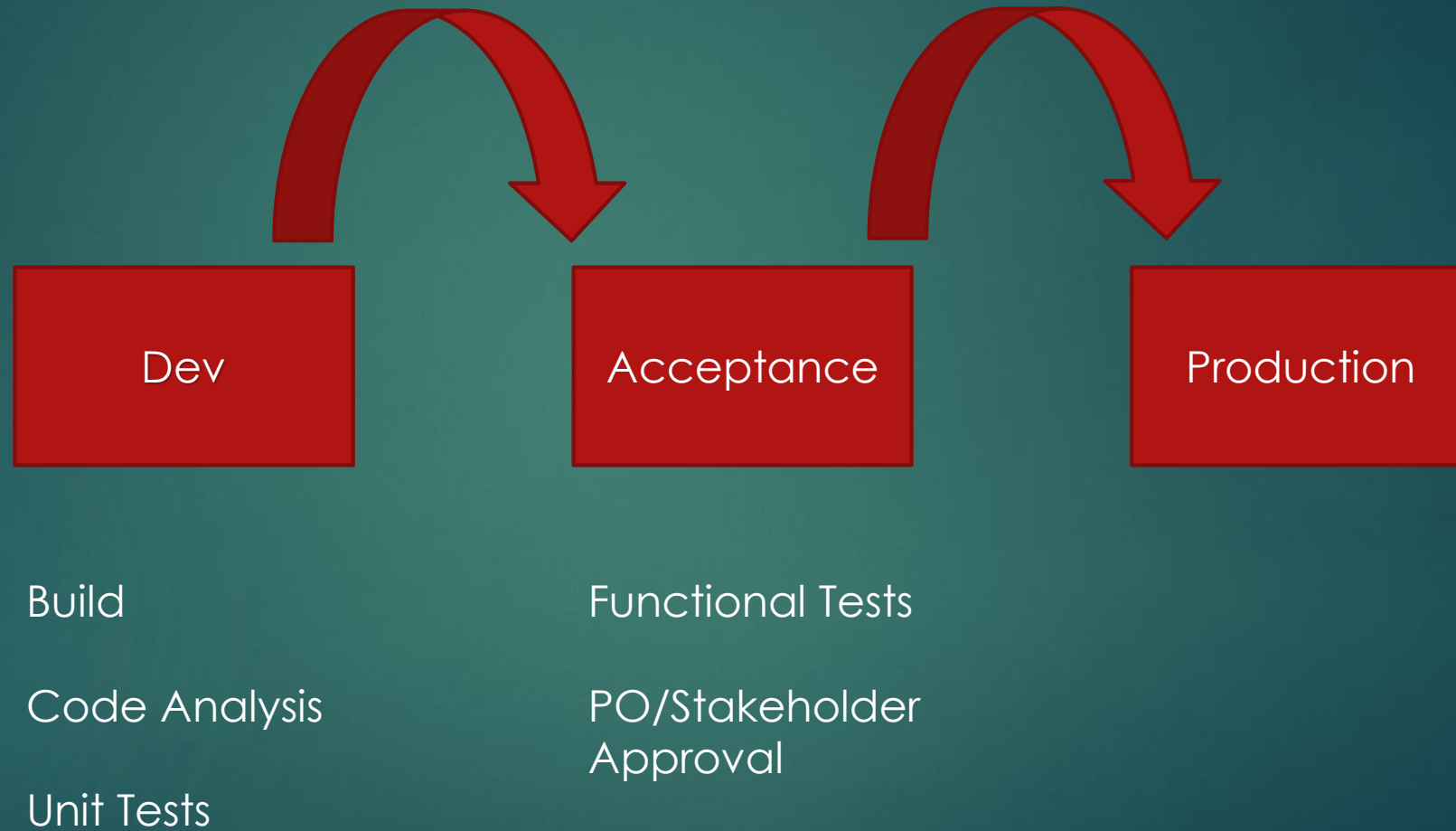
Sample from winnipeg



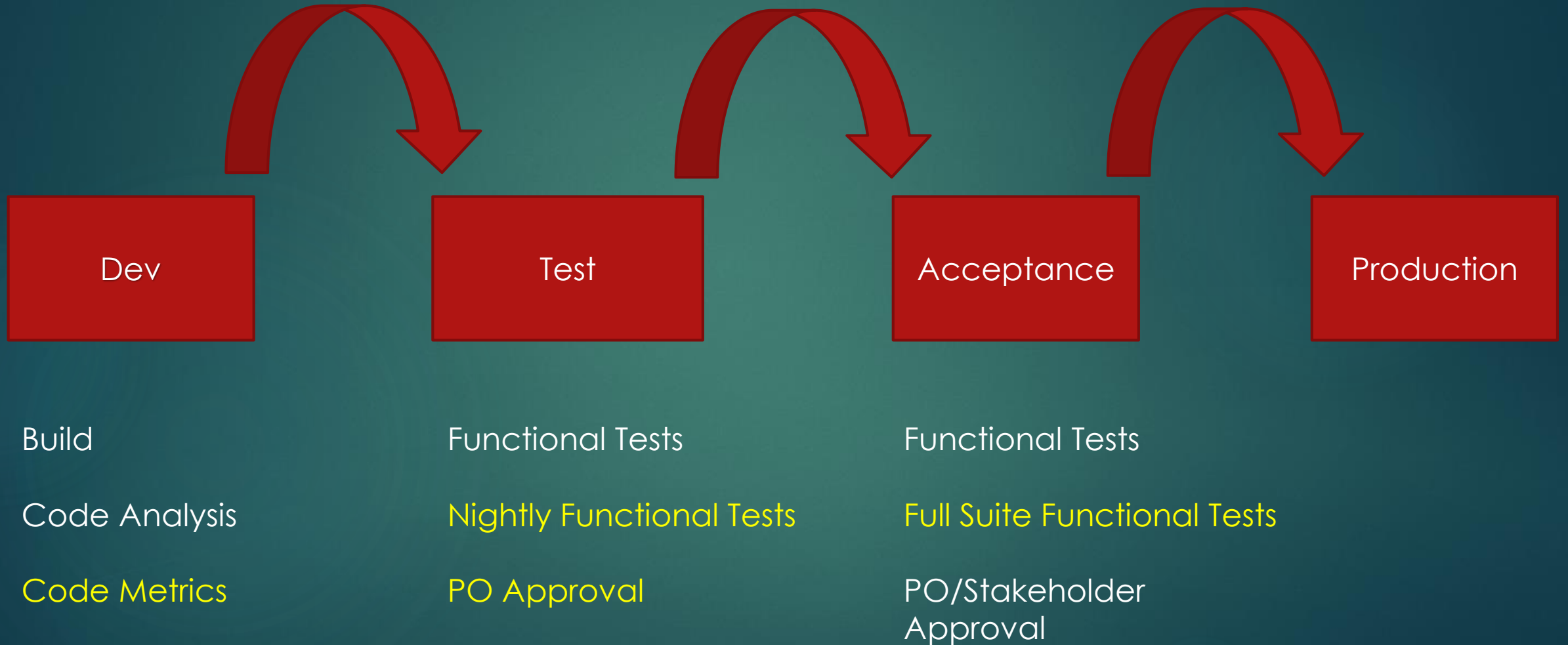
Our Story Map

	Code Quality		Tests		Environments		
	Static Code Analysis	Code Metrics	Unit Tests (UT)	Functional Tests (FT)	Test /QA	Acceptance	Production
Release 1	Setup SonarQube Run Code Analysis		Execute UT Fail build if UT fails	Execute FTs Fail build if FT fails		Setup Deploy Config Mgmt	Setup Deploy Config Mgmt
Release 2	Setup Custom ruleset	Calculate Code Metrics on Solution	Fail build if overall coverage is < 85%	Nightly Full Regression Email Report	Setup Deploy Config Mgmt		
Release 3	Email build result notifications	Fail Build if Metrics are less than threshold	Fail build if new coverage is < previous coverage	Fail build if coverage is < 85%			
	Code Quality result notifications		Performance Tests Security Tests Database Unit Tests Api Testing		Infra as Code Auto promotion based on test results		

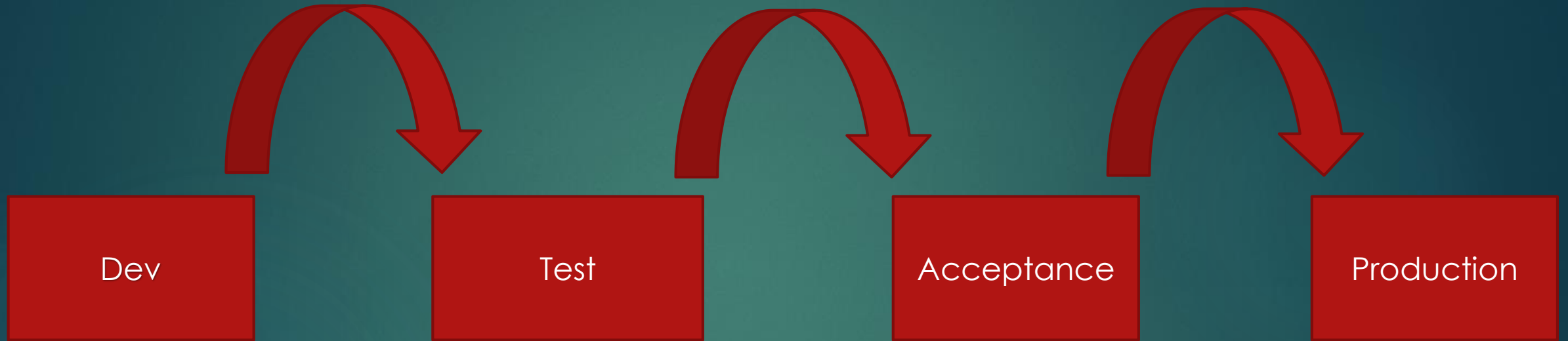
Release 1



Release 2



Release 3



Build
Code Analysis
Code Metrics
Code Coverage

Functional Tests
Nightly Functional Tests
PO Approval
Functional Test Coverage

Functional Tests
Full Suite Functional Tests
PO/Stakeholder Approval

Takeaways

- ▶ Positives
 - ▶ Story mapping
 - ▶ Early value realization

- ▶ Improvements
 - ▶ Limit tool stack
 - ▶ More POC's because of variety of tech stack

Second Application

- ▶ Quality Gates
- ▶ Product Quality
- ▶ QA, Acceptance, Production environments

- ▶ Monolithic
- ▶ Many configurable components
- ▶ Multiple customers
- ▶ Multiple versions

Story Map

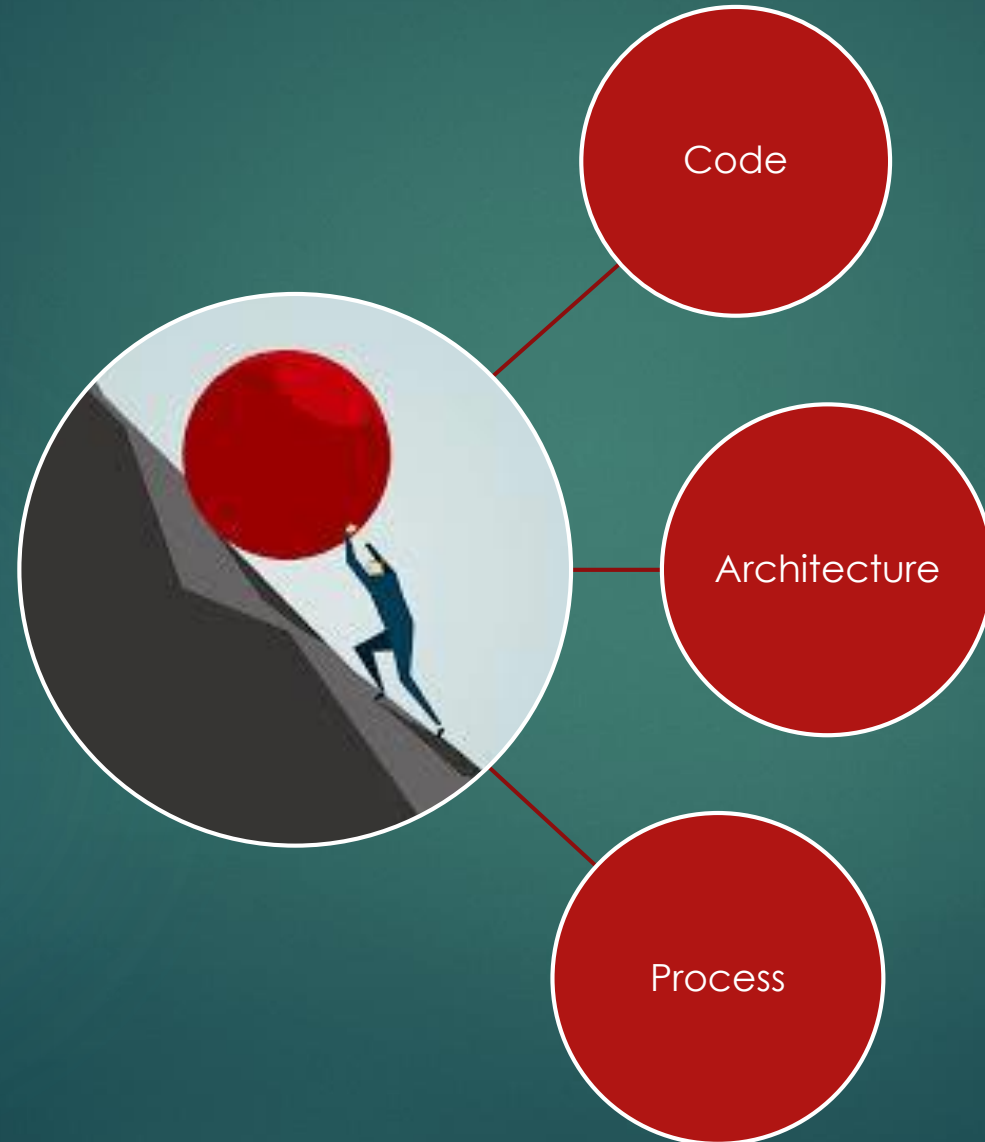


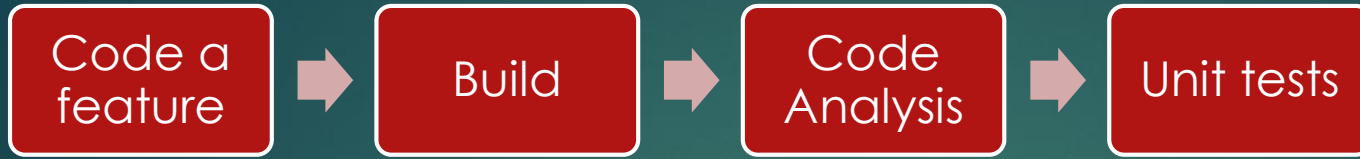
	Configurable Components		Customers		Version Support		
	Module Specific	System wide	Customer A	Customer B	Version 4.0	Version 5.0	Version 6.0
Release 1							
Release 2							
Release 3							

Challenges

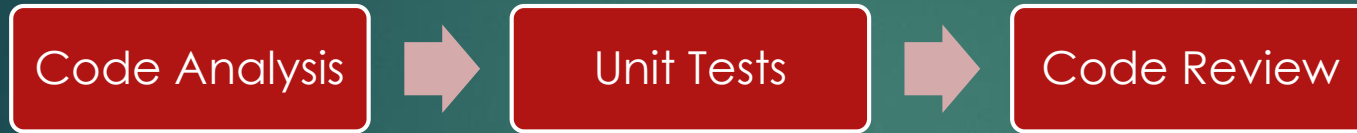
- ▶ Time to understand the entire system
- ▶ Feedback cycle is longer
- ▶ Identifying tools
- ▶ Collaboration with IT

Before getting started..





Pull Request



Merge



Takeaways

- ▶ Positives
 - ▶ CAP worked well.
 - ▶ Break Monolithic application into smaller applications
 - ▶ Involvement of Dev from scratch instead of KT at the end.
- ▶ Learning
 - ▶ Show value early
 - ▶ Collaboration with IT/Network team
 - ▶ Changes in Branching strategy

CD in 90 days

- ▶ Goal driven
- ▶ Emphasis from stakeholders
- ▶ Focus

Thank you