

2014 IEEE  
Software Technology Conference  
Long Beach, CA



BUREAU OF THE  
**Fiscal Service**  
U.S. DEPARTMENT OF THE TREASURY

Treasury Offset Program, Department of the Treasury –  
Bureau of the Fiscal Service

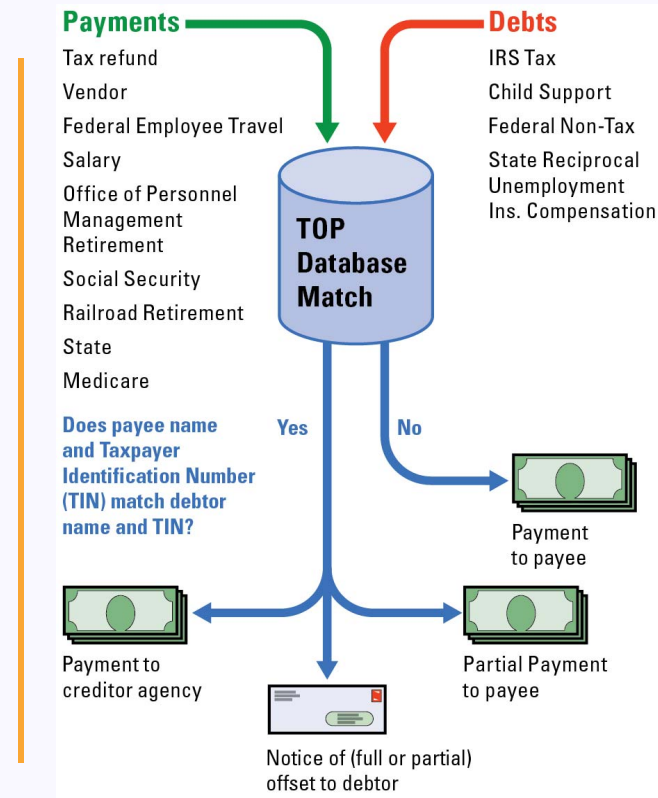
Alyssa Riedl, Director

**Transforming Your IT Project in the Cloud**



# What is the Treasury Offset Program?

- TOP is a centralized offset process that intercepts Federal payments of payees who owe delinquent debts to agencies that have submitted debt information to Bureau of the Fiscal Service.
- *Offset* is withholding funds payable by the United States to a person to satisfy a debt owed to the United States or to a state.
- *Centralized offset* is the offset of payments disbursed by Bureau of the Fiscal Service and other Federal disbursing agencies through the TOP.



# TOP NG Background

- Debt Management Services (DMS) is building a system that will replace the current Treasury Offset Program system. Why?
  - TOP is a prototype that evolved over the years through patches
  - TOP is approximately 15 years old and written using a combination of COBOL programs and Java (the TOP web client)
  - TOP has reached capacity in processing payments during peak periods
  - TOP lacks flexibility to accommodate changes in the system
- The goal of this project is to increase TOP collections by enabling:
  - Increases in payment streams
  - Increases in debt volume
  - Matching effectiveness

# TOP NG and Agile / Lean Principles

- The First Principle of Agile
- Principles drive the practices
- Practices used by TOP NG team:
  - Collaboration & Instant Feedback
  - Requirements & Code
  - Defect Management
  - Continuous Integration & Automation
  - Automated Deployments
  - Code Quality
  - Definition of Done

## First Principle of Agile

Our highest priority is to satisfy the customer through early and frequent delivery of valuable software.

Supporting this principle drives the decisions and practices implemented by the team.

# Agile / Lean Practices Used By TOP NG Team

## Kanban Practices

- Visualize the work
- Limit Work In Progress (WIP)
- Manage Flow
- Evolutionary Change
- Improve Collaboratively
- Evolve Experimentally
- Stand up focusing on Stories rather than individuals

## Scrum Practices

- Product Backlog
- Stand up meeting
- Retrospectives

## XP Practices

- Acceptance Test Driven Development
- Pair Programming
- Refactoring
- Continuous Integration
- Collective Code Ownership
- Simple Design
- Sustainable Pace
- User Stories
- Collaborative Work Space
- On-site Customer

## TOP NG Perspective on the Cloud

- The cloud meets expectations
  - Lead time to get servers is a “thing of the past”
  - Using the cloud is seamless
  - Cloud resources are relatively inexpensive
  - Less reliance on traditional support groups

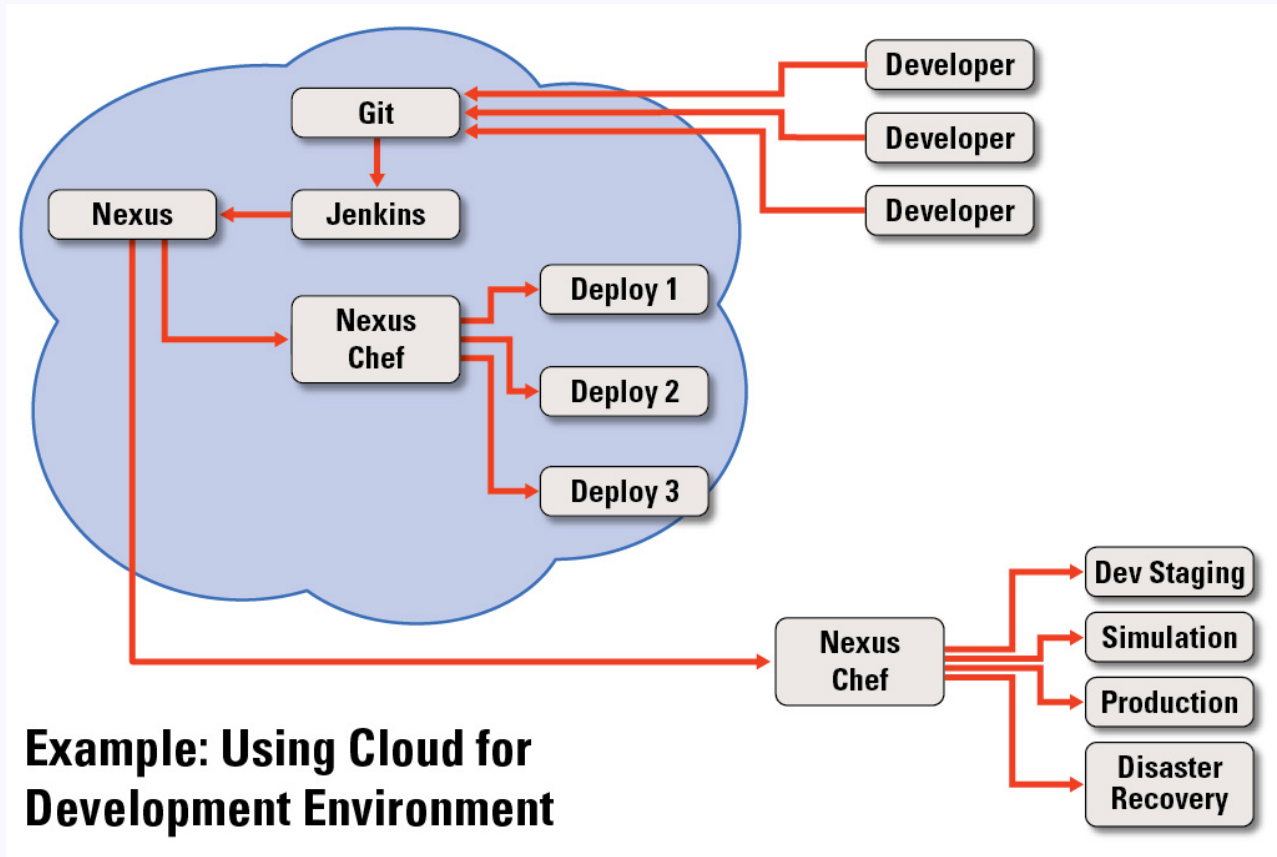
# Benefits of Using Cloud for Development

- Cost and Schedule Friendly:
  - Setup development environment & obtain new server instances in minutes
  - Rapid prototyping of new architectures
  - Shortens long hardware/software procurement processes
  - Quickly scale capacity up and down as computing requirements change
  - Only pay for capacity that you use
- Provides an environment that allows for close monitoring and tuning of the system for maximum performance AND for agile operations and maintenance.



# Development Cloud Best Practices

- Randomize your data – do not use production data in the cloud
- Migrate existing support resources with you to the cloud (i.e. DBA, system administrator, etc.)
- Leverage virtualization for productivity
- Understand your FISMA rating and document your project's controls
- Database As A Service: Amazon RDS



# Cloud Benefits for TOP NG Program

No Up-Front  
Capital Expense



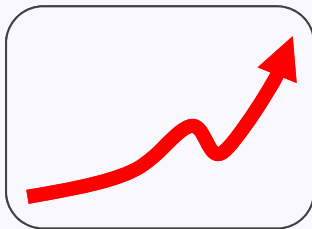
Low Cost



Automated, Self-  
Service Infrastructure



Easily Scale  
Up and Down



Pay Only for  
What You Use



Improve Agility & Time-to-  
Market / Mission



## Collaboration and Instant Feedback

- Constant collaboration is key to quality requirements and software; It eliminates waste for faster delivery of better value.
- Turning to a teammate and asking a question that gets immediately answered and validated by working on something together results quickly and accurately in a value-added feature.
- Teams should be organized without silos: product owners, developers and quality assurance all sitting side by side during the project's development.
- Hand-offs between team members, where someone talks and someone listens, have a cost:
  - Email and hand-offs can cause delays and re-work for complex topics
  - A tester might interpret the requirements one way; a developer might interpret the requirements another way; and none of those ways is what the requirements analyst meant

## Benefits Gained from Collaboration

- High-quality software delivered as advertised
- Greater transparency for stakeholders
- Eliminates waste for faster delivery
- No backlog of defects or bugs
- Simplicity due to evidence-based decisions
- Evolving cross-functional team
- Better environment for innovation and learning

# Requirements and Code -1

## Acceptance Test Driven Development (ATDD)

### 1. Requirements are written as executable test scenarios:

- Created during requirements elaboration and before any code is written
- Written collaboratively with developers, product representatives and tester
- Documented as readable files, in business language, and can be shared with stakeholders
- Examples of how the system should behave under specific conditions
- When executed successfully, we know the desired feature has been correctly built

## Requirements and Code -2

### Acceptance Test Driven Development (ATDD)

2. Application code is written to execute the tests:
  - Scenarios become part of the code base, so the requirements are always up to date
  - Code is written to execute the tests and assert & verify expected results
  - These tests drive code development (keeping it simple)

# Code Quality

TOP NG is using Acceptance Test Driven Development

TOP NG measures:

- Testability
- Security
- Performance
- Code coverage
- Copy and pasted code
- Adherence to best coding practices
- Complexity
  - Real-time feedback in developer environment
  - Trending measured on build server



## Defects: Traditional versus Agile / Lean

### Traditional Development Model:

- Developers build a feature and hand it off to QA to be tested
- The QA team tests the feature and discovers a defect
- The defect is logged, prioritized, and at some future date, addressed by the development team
  - Delays due to departmental and hand-off/communication issues AND developers have moved on to new features
- At some point after all known “SEV 1” defects are fixed, a decision is made to go into production
- The remaining “low priority” defects may or may not ever get addressed

## Defects: Traditional versus Agile / Lean

### Agile Development Model on the Cloud:

- Developers and QA work closely together to define acceptance criteria before development begins.
- They communicate at least daily and whenever questions or issues arise.
- If QA discovers a problem, it is immediately fixed by the developers before the story is accepted.
  - There is no need to log or prioritize the defect.
- At some point, a decision is made to go into production. There are no known defects.

In either model, defects will inevitably be discovered in production.

**Agile / Lean process results in higher-quality software, with no known defects.**

# Continuous Integration

- Every time code is checked in:
  - Build all the deployable artifacts
  - Execute ~3,600 unit, integration and customer automated tests against H2 and Oracle databases
- Nightly
  - Generate quality metrics via tools like Sonar & CAST
  - Execute and validate conversion process
  - Execute performance test and ensure it is within threshold
- On-Demand via a button click
  - Deploy to the cloud acceptance servers (6 minutes)
  - Deploy to Treasury data center (10 minutes)

# Automated Deployment

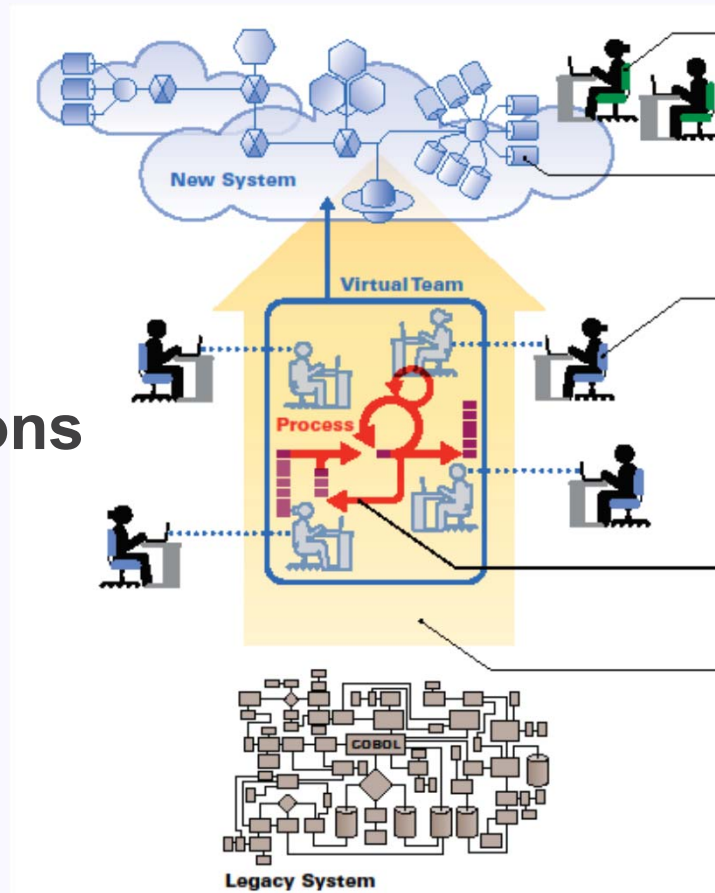
Configuration tools like Chef are used to manage server configuration and application deployment.

Server configuration is managed side-by-side with code and can follow the same configuration management process.

Configurations can be tested in the cloud before being fully deployed.

# Summary

# Multiple Transformations



New Management Approach

Apply New Technology

New Development Team Concepts

Pilot Use of New Methodologies

Take a Greenfield Approach to New Solution



## Summary and Questions

- Principles drive practices and decisions
- There are no best practices, only practices that work best for your team
  - Questions?

# Thank you!

Department of Treasury – Bureau of Fiscal Service

Steven Kennedy

Program Manager

Debt Management Services

Bureau of the Fiscal Service

U.S. Department of the Treasury

[Email: Steven.Kennedy@fiscal.treasury.gov](mailto:Steven.Kennedy@fiscal.treasury.gov)



BUREAU OF THE  
**Fiscal Service**  
U.S. DEPARTMENT OF THE TREASURY



# Acronyms

ATDD – Acceptance Test Driven Development

DMS – Debt Management Service

IT – Information Technology

QA – Quality Assurance

TOP – Treasury Offset Program

TOP NG – TOP Next Generation

WIP – Work in Progress