Overview

Background Information
- RIT's Information & Technology Services
- Enterprise Support, Operations, and Applications Development
- 40K – 45K inbound phone calls per year
- 15K – 20K inbound emails per year

Motivation
- Facilitate ITS Service Desk operations
- Identify abnormal and outstanding performance
- Show what Service Desk does in terms of workload
- Pre-existing dashboard lacks historical information and extensibility

Requirements

Functional
- ITS Systems data aggregation for short-term storage, processing, and metric visibility
- Querying API
- Views
  - Public Dashboard
  - Service Desk TV Dashboard

Non-Functional
- Extensibility
- Modular, plugin-based design
- Testability
- Test-Driven Development
- Performance
- Security

Design

Subsystem Interaction
- Client
  - Web-Based Views
- Server
  - Front Controllers
  - Web API
  - Domain Entry Points
  - Domain Layer
  - Web Core / Domain Logic

Data Aggregation System

Design to Schedule

Planning and Risk Management
- Tuesday Sponsor Meeting
- Thursday Process and Risk Management Meeting
- Saturday Standup Meeting

Test Driven Development

Outcomes

Public Dashboard

Service Desk Dashboard

Future Work
- Service Desk Admin View
- Widget-based Service Desk Dashboard View
- Ticketing System Plugin
- Push notifications
- Additional data sources and views

Lessons Learned
- TDD for web projects can be difficult for some areas but very useful in others
- A more agile approach would allow for more opportunities for sponsor feedback
- Design to schedule still worked well for us because of the much needed upfront requirements gathering and elicitation phase