

# Project Plan

## Overview

The Two Sigma senior project is to create a data ontology system to discover and exploit relationships between artifacts and data gathered from disparate sources to enable Two Sigma to develop better business insights from existing data and systems. The system consists of several components: the collector API, ontology cache, query and reporting engine, command line interface, web interface, and a security manager.

The system will run on the Java Virtual Machine. Most of the components in the system will be implemented in Java 6, with scripting elements implemented in Groovy. Several existing libraries and components are being considered for use in the system, the main one being Jena, a Java library used for building and querying ontologies. We are entertaining options for the persistent data store that will store the ontology cache. The web interface will be implemented using Google Web Toolkit.

The customer for this project is Two Sigma. This is a project that will be used internally to analyze artifacts that Two Sigma currently stores in disparate systems. As a result of the exploratory nature of this project, the schedule and process described below are driven by a prioritized set of risks.

## Goals and scope

### In scope

1. Collector API
2. Ontology model
  - Versioning
3. Instance data cache
  - Versioning
4. Scripting engine
  - Query interface
5. Command line utilities

- Ontology editor
- Query engine
- 6. Web interface
  - Ontology editor
  - Query engine
  - Report generation
  - Graph browser
  - Conflict resolver
- 7. Security manager
- 8. Sample collectors

## **Out of scope**

1. Collectors for actual TwoSigma systems

## **Deliverables**

All documents are "living" and will be changed and refined as the project progresses.

First quarter:

1. Project plan
2. Software Requirements Specification (SRS)
3. Testing plan
4. High-level architecture diagrams
5. Risk mitigation plan
6. Architecture baseline implementation

Second quarter:

1. Refined versions of documents from first quarter
2. Javadocs for final implementation
3. Final system implementation
4. Sample collectors
5. Project postmortem
6. Project poster

## Risks

### Requirements Risks:

1. Requirements are vague and defining details later in the schedule expands the project scope past required deadlines
2. Feature creep
3. Missing deadlines in a tight schedule (overly optimistic schedule)

### Technological Risks:

1. Gaining an understanding of the ontology technology
2. Interacting with TwoSigma's domain-specific data through the Collector API and the query interface
3. Inadequate test cases and/or coverage

## Scheduling and Estimates

The schedule is divided into two quarters, one for research, requirements and ground work, the following focusing on features. Goals of the first quarter are as follows:

- Research required technologies, determine if technology meets the needs of the project and learn basics of the technologies to better refine requirements and design system
- Begin architectural design
- Implementation to be delivered:
  - Basic collector API allowing for pushing of data into the cache
  - Basic ontology definition loading
  - Infer relations between data from ontology
  - Persist instance data
  - Basic security modeling and verification
- Complete requirements specification and get Two Sigma approval for features to be implemented in the first iteration of second quarter

The schedule for the remainder of the first quarter is as follows:

- Week 3-4: Technology spike, researching, learning technologies, and gathering and documenting requirements for iteration during weeks 5-8. Technology spikes are managed via Trac tickets in the "Tech Spike 1" Milestone.
- Week 5-8: Design, architecture, initial implementation and testing of ground work defined above (*Increment 1*)
- Week 9-10: Review of product with Two Sigma, planning, refactoring for second quarter (*Increment 2*)

The second quarter will focus on implementation of the remaining features, notably the following:

- Web Interface, with user control (security), advanced/simple user query interfaces
- Security model
- Script query engine
- Collector API with support for security/source metadata
- Command line interface
- Sample collectors
- Versioning of ontology and data cache

Estimates for individual features for the second quarter will be determined at the end of the first quarter. This estimation is dependent on knowledge gained from requirements elicitation and technology development activities.

## **Measurements & Metrics**

First quarter metrics:

1. Requirements volatility (measuring feature creep)
2. Risk volatility (measuring change in prioritization)
3. Hours worked (measures project velocity)

Second quarter metrics to be determined at the end of first quarter.

## Technical Process

First quarter process

- Evolutionary prototyping
- Two iterations for the first quarter
- Use prioritized risks in order to drive project activities
- 

The second quarter will consist of a more timeboxed series of iterations, with focus on mitigating risky features. As we get closer and perform research and risk mitigation, we will know more about the nature of the 2nd quarter process.