WPI Suite TNG

Advisor: Prof. Gary F. Pollice

Core	*Exemplar*
Michael Della Donna	Christopher Casola
Brian Gaffey	Andrew Hurle
Ryan Hamer	Jennifer Page
Tyler Wack	

Problem

- A project for software engineering students
- Lack of tools for software engineering students

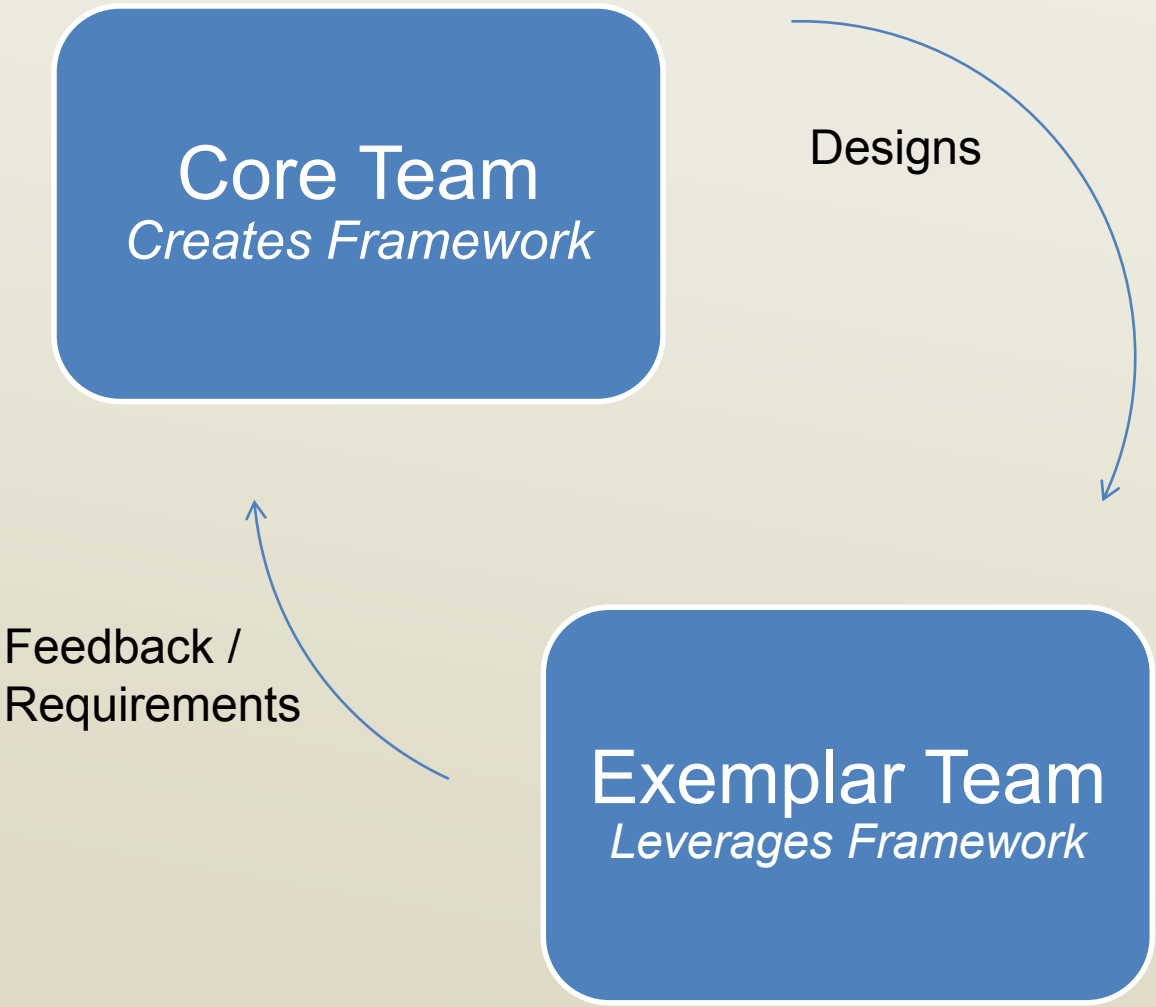
Previous Solution

- Mainly written by students
- Lacked comprehensive unit tests
- Performance issues
- Lack of Security
- No Client-Server architecture

Goals

- Well-defined module framework
- Configurable
- Client-server
- Excellent documentation
- Exemplar module(s) with documentation
- Open source availability

Approach


Core Team
Creates Framework
Designs
Exemplar Team
Leverages Framework
Feedback /
Requirements

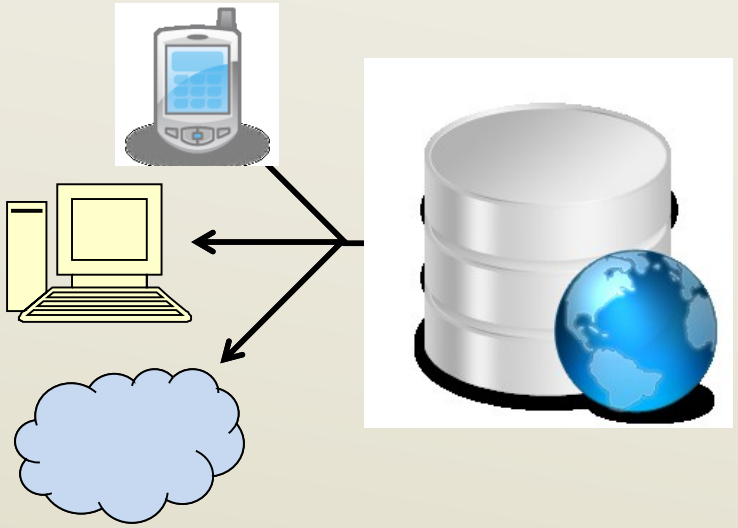
Core

Goals

- Flexible
- Avoid Complexity
- Comprehensive Documentation
- Robust and Maintainable

Flexible

- Client-Server architecture
  - Client agnostic
- Module System
- HTTP REST API
  - Using JSON for communication

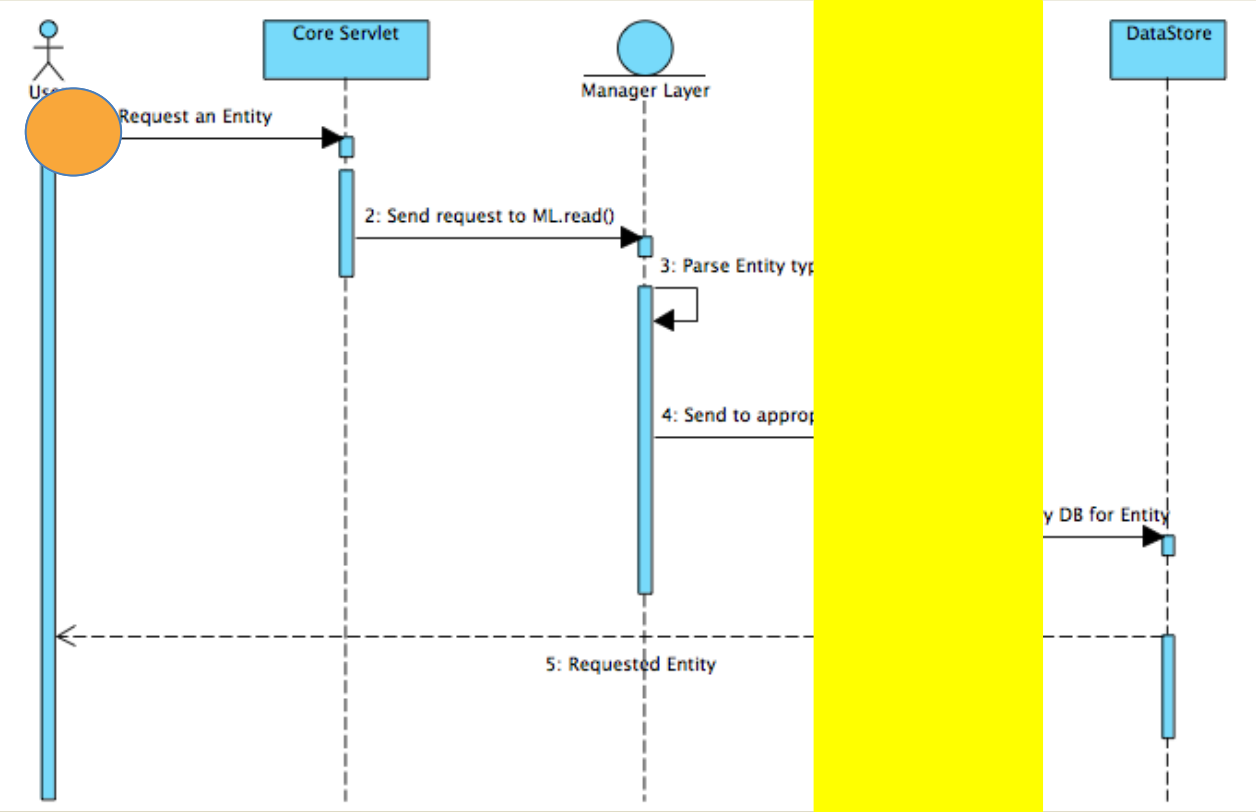


Avoid Complexity

- Abstraction
  - Network Communication
  - Persistent Storage
  - Login Security

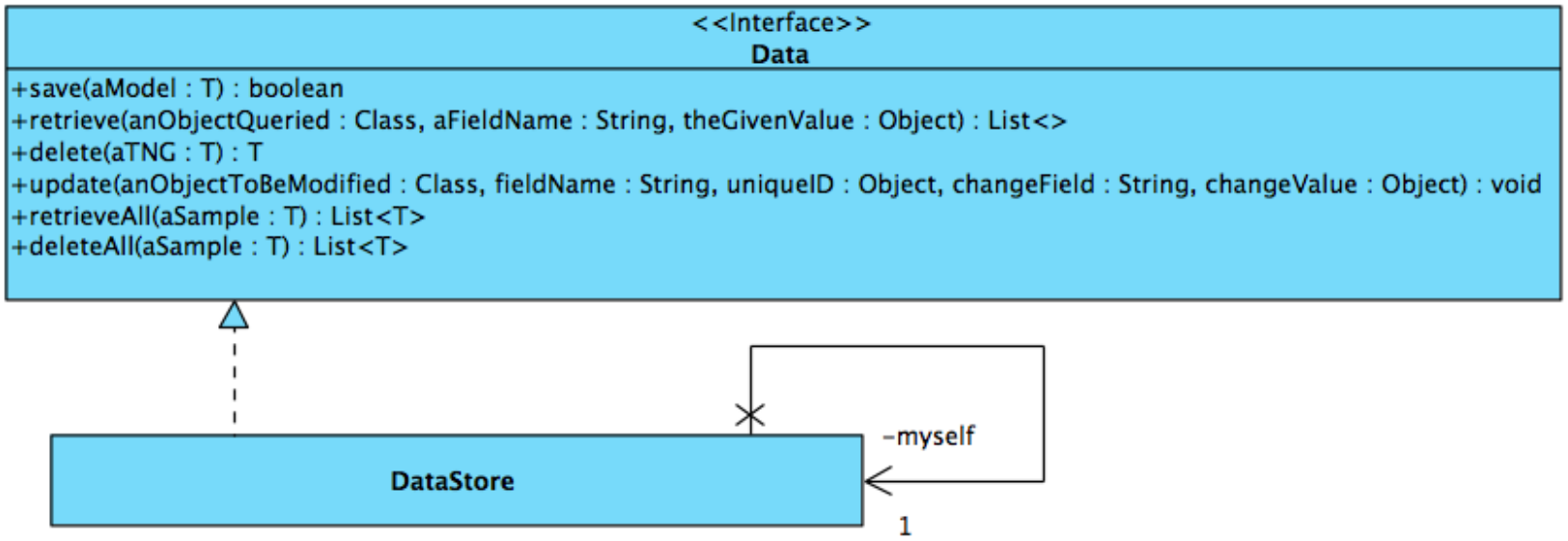
Avoid Complexity

Network Communication



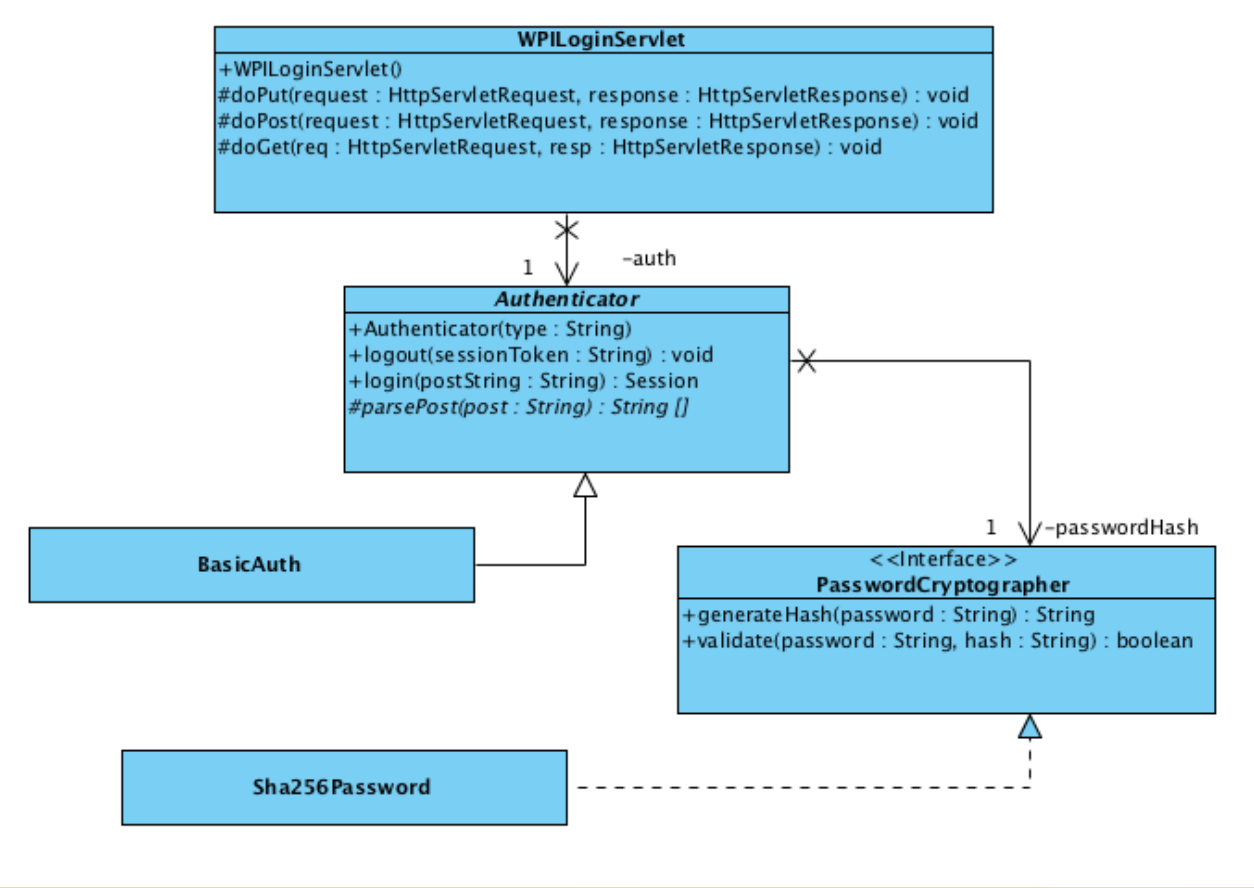
Avoid Complexity

Persistent Storage Interface



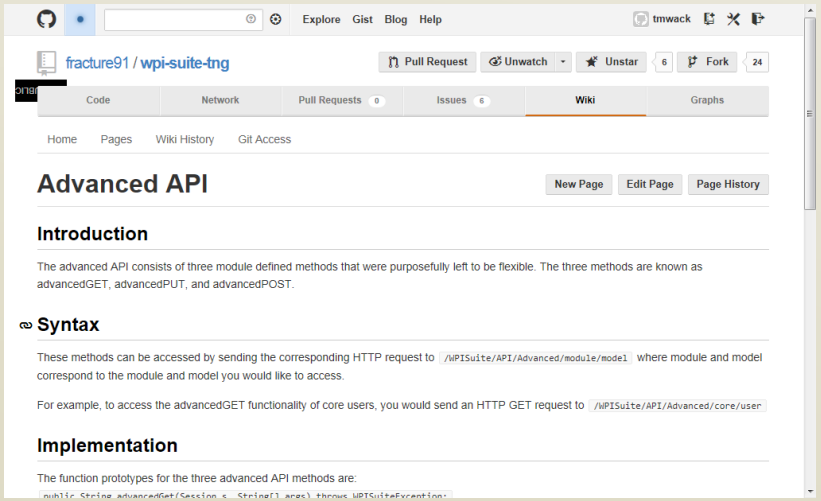
Avoid Complexity

Login Security



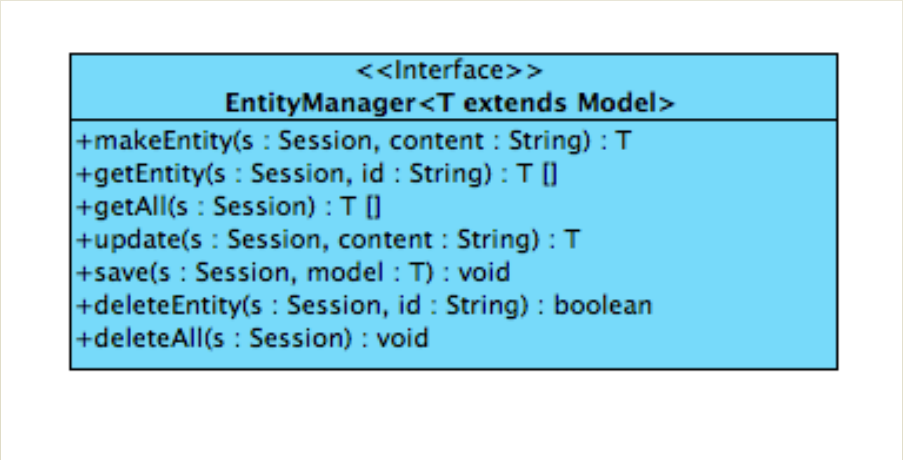
Comprehensive Documentation

- Well-documented components
  - Apache Tomcat
  - Db4o
  - Google GSON
- As well as maintaining our own

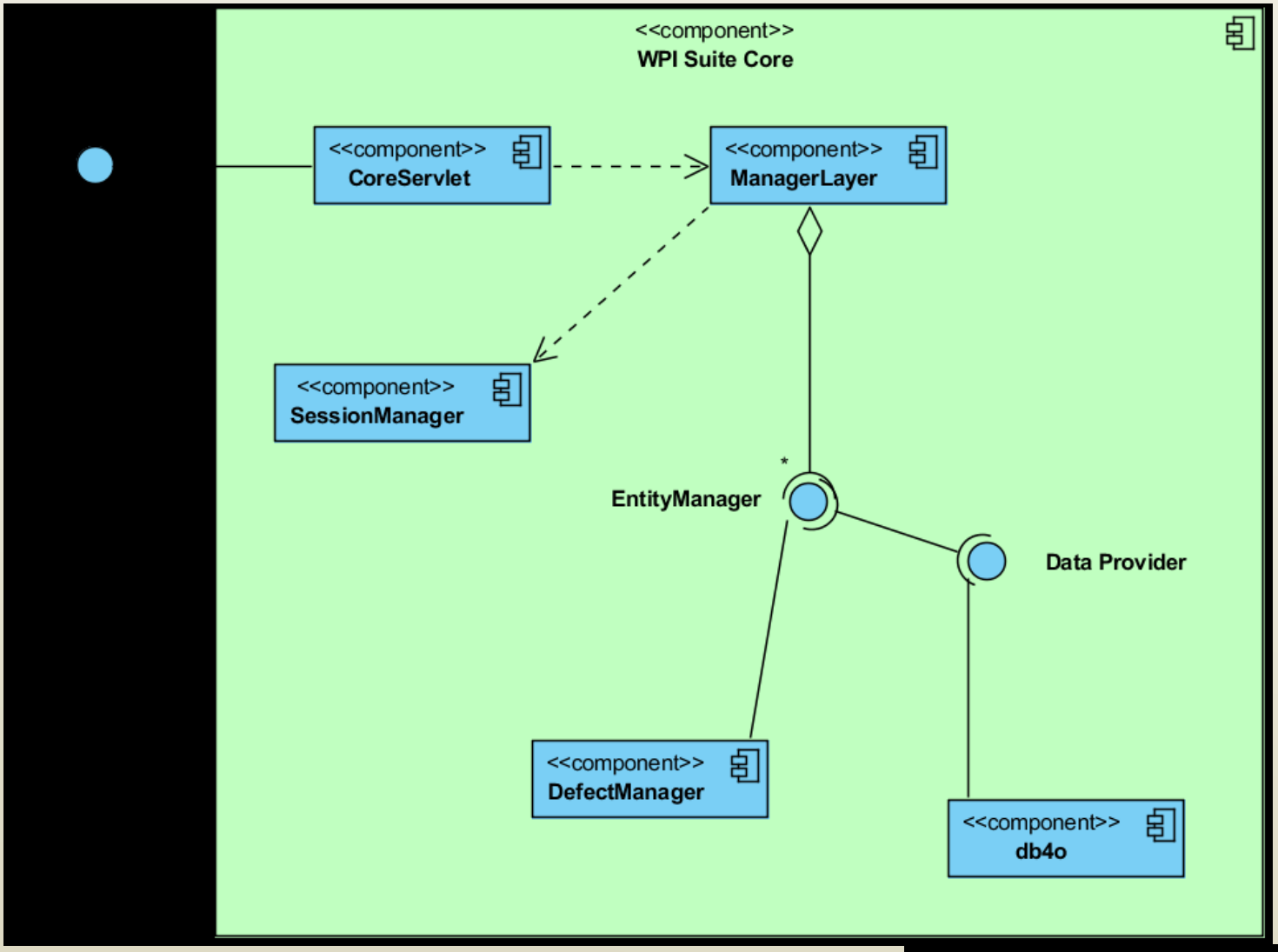


Robustness

- Creating a module only requires 7 methods
- Descriptive Error Handling
- Server side logging



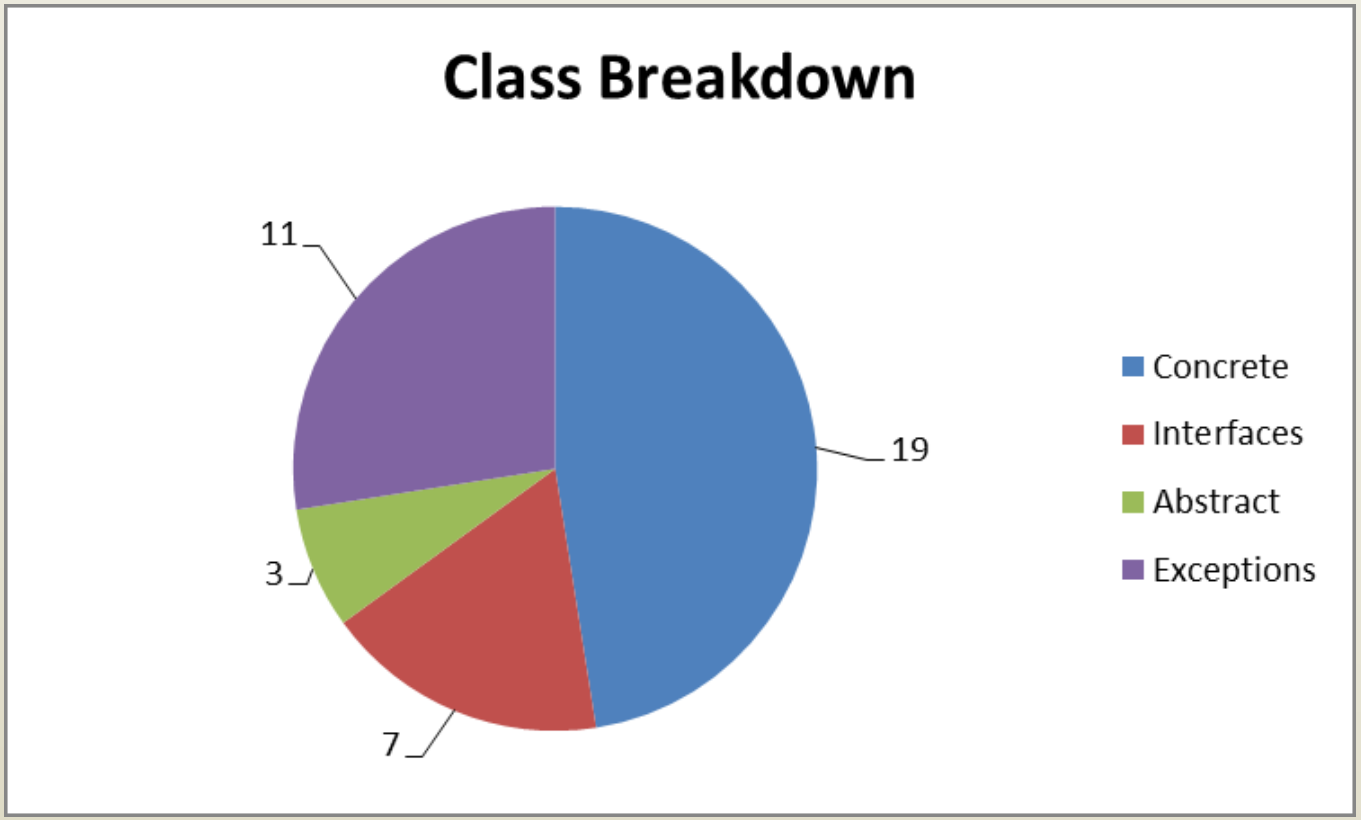
Architecture Diagram



Code Metrics

- 6,192 lines of code
- 52% code coverage
- 37.3% comment ratio
- 40 Classes

Code Metrics



Looking Forward

- Dynamic Module Loading
- Increased Test Coverage
- Stronger Web Admin Console

Exemplar Module

Defect Tracker

Goals

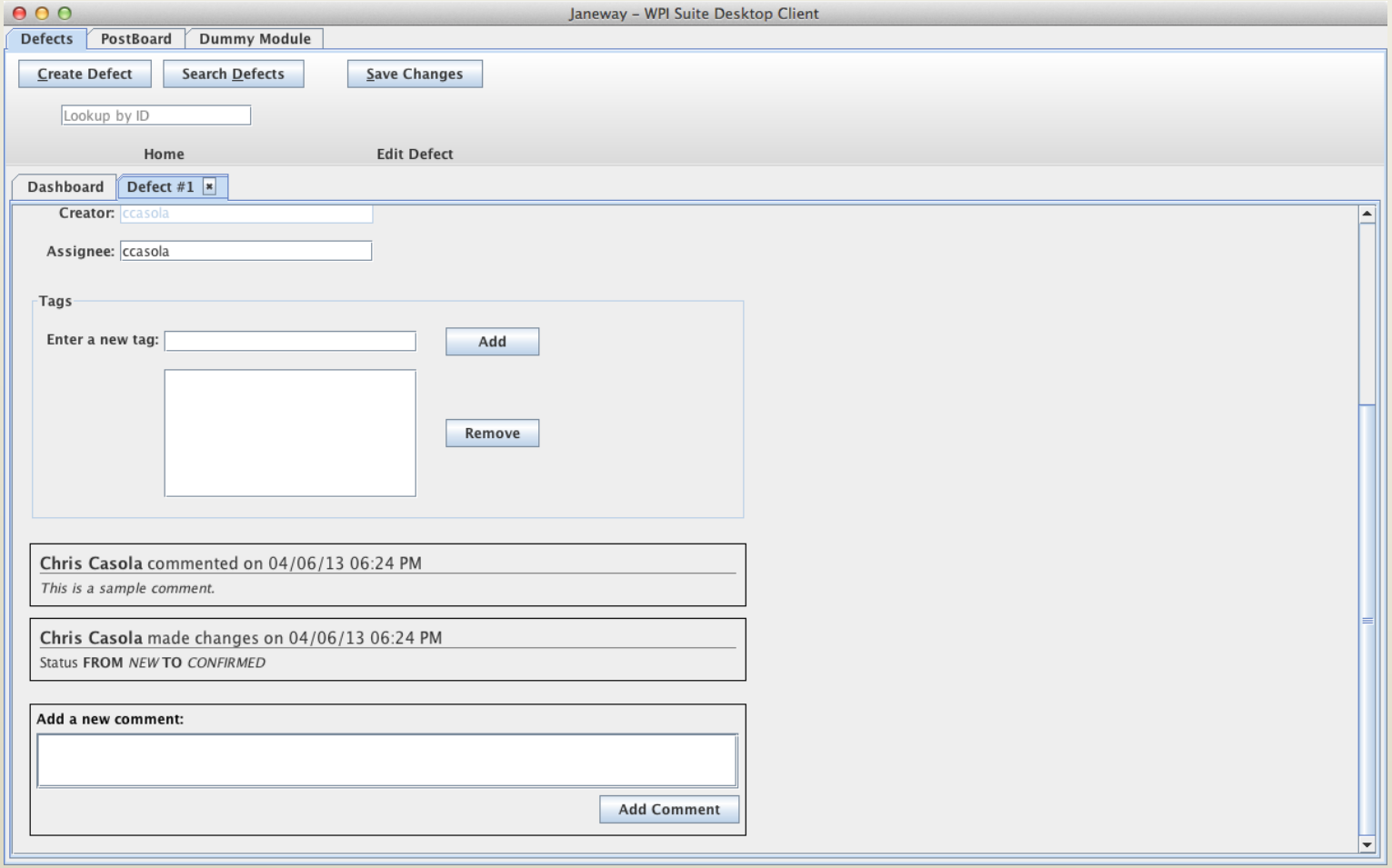
- Extensible desktop client
- Example module for students to learn from
- Simple network communication
- Good documentation and design

Design Decisions

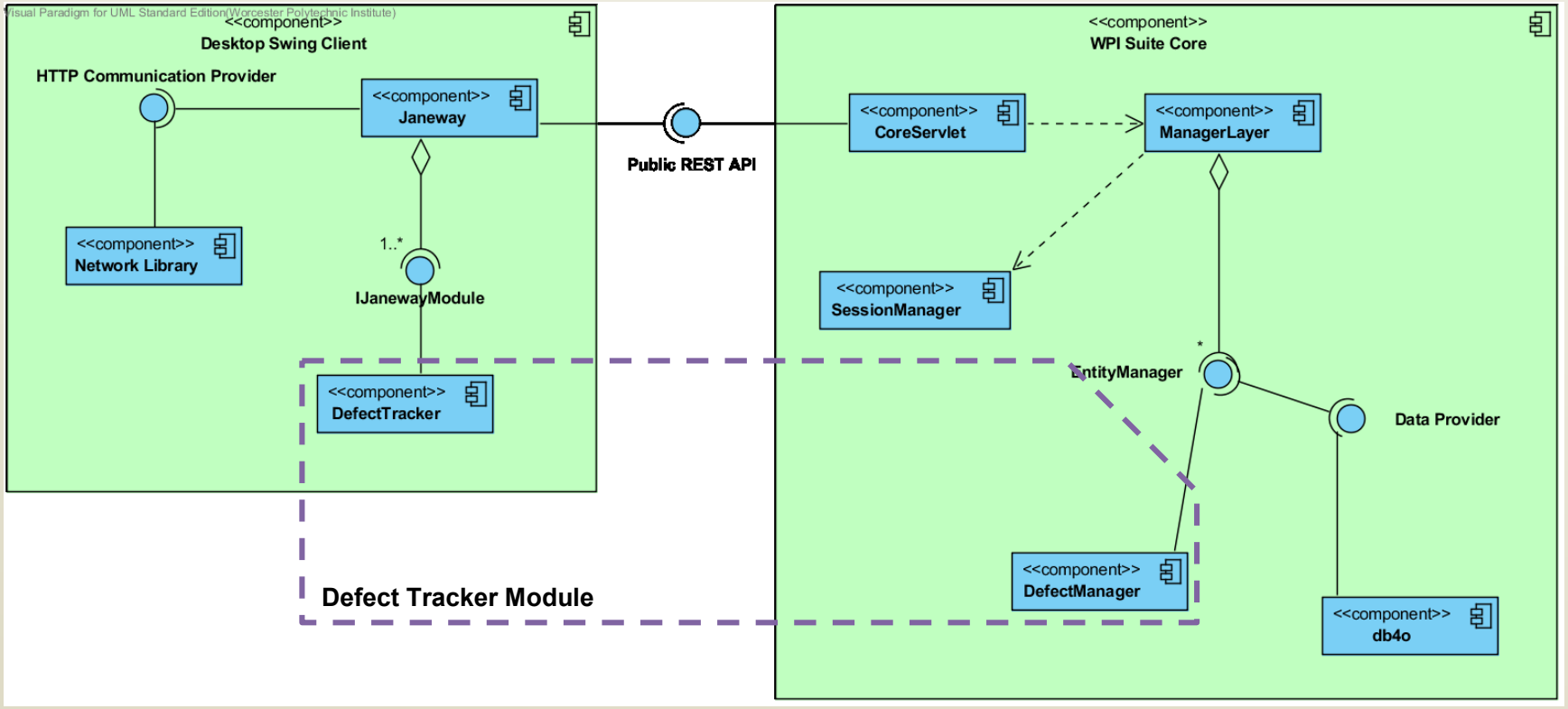
- Defect tracking
- Java and Swing
- Dynamic module loading
- Modules provide tabs and toolbars
- Cross-platform look & feel
- Modeless editing
- Network library

Desktop Client

Desktop Client



Architecture



Code Base Metrics

- 6,836 SLOC
- 17.6% Comment Ratio
- 150 classes
- 28.1% Testing Coverage
  - Server-side code: 93%
  - Network library: 70%

Documentation

Creating an *EntityManager*

To allow our PostBoard module to save message in the core we must provide an entity manager for PostBoardMessages. The first step is to create a class that implements the *EntityManager* interface. Each EntityManager is responsible for all requests involving one Model class. The core automatically forwards all requests to the EntityManager. An EntityManager implements the generic `EntityManager<T>` interface.

We will start by creating a new class with the following signature:

```
public class PostBoardEntityManager implements EntityManager<PostBoardMessage>
```

As shown, you must provide the type of the object that this EntityManager will be storing (PostBoardMessage). Eclipse should also automatically generate method signatures for all of the methods required by the EntityManager interface.

You need to provide a simple constructor for the entity manager that takes as an argument a reference to the database so that it can be saved in a field.

```
public PostBoardEntityManager(Data db) {
        this.db = db;
}
```

Next, we need to implement the `makeEntity()` method. This method is responsible for receiving a message in JSON form, parsing it into an actual PostBoardMessage, and then saving it in the database. It must also return the PostBoardMessage that was saved back to the client. It is called whenever a PUT request is received to /postboard/postboardmessage.

```
public PostBoardMessage makeEntity(Session s, String content)
                throws BadRequestException, ConflictException, WPISuiteException {

        // Parse the message from JSON
        final PostBoardMessage newMessage = PostBoardMessage.fromJSON(content);

        // Save the message in the database if possible, otherwise throw an exception
        // We want the message to be associated with the project the user logged in to
        if (!db.save(newMessage, s.getProject())) {
                throw new WPISuiteException();
        }

        // Return the newly created message (this gets passed back to the client)
        return newMessage;
}
```

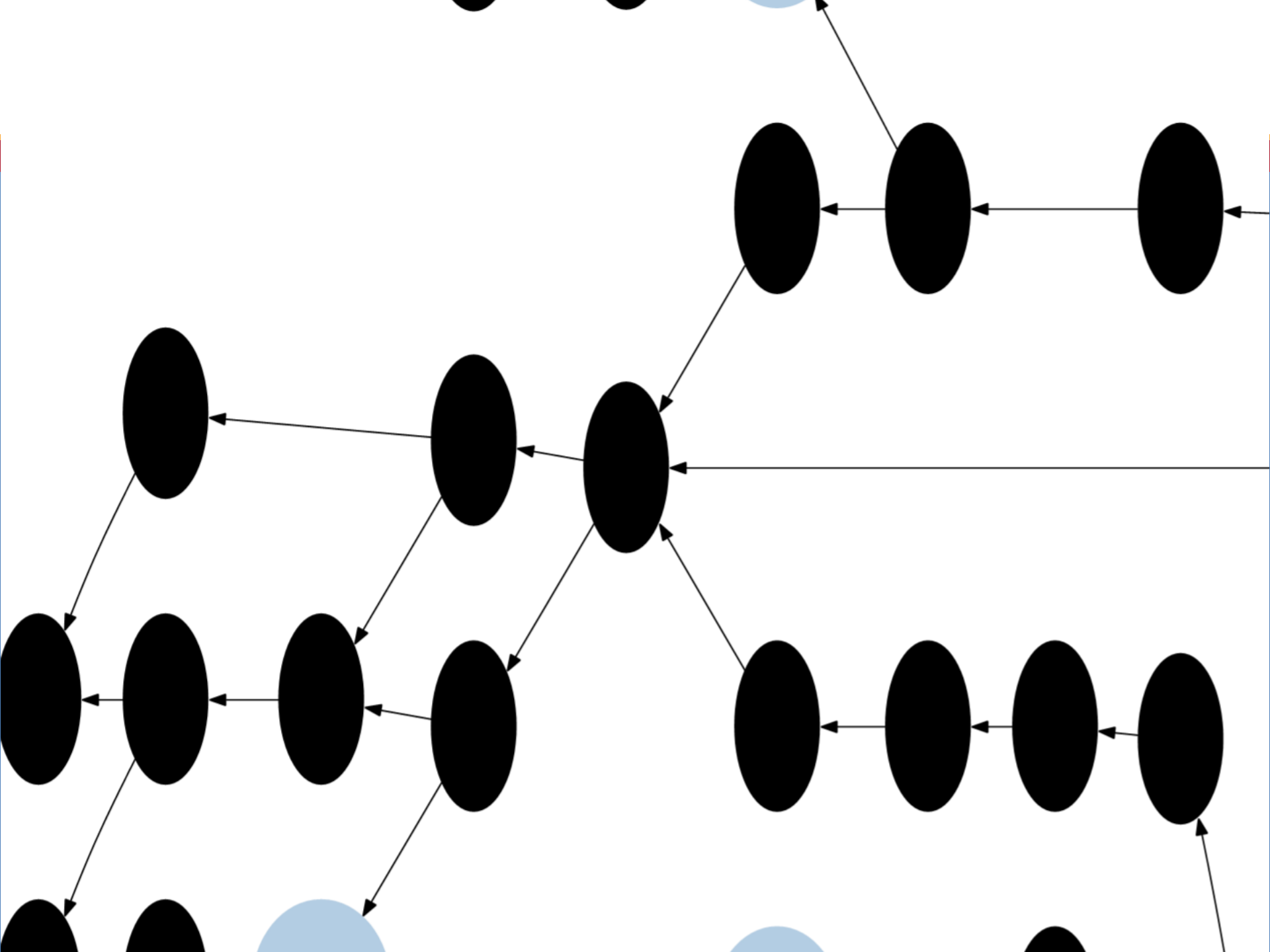
Lets walk through the code above step-by-step. The first parameter of makeEntity is a Session object containing information about the user making the request. The second parameter is a String containing the body of the HTTP request (in this case the body contains a JSON-encoded PostBoardMessage). The first line of code in the method uses the static method PostBoardMessage.fromJSON to convert the JSON content into a new PostBoardMessage. The if block attempts to save the PostBoardMessage in the database, throwing an exception if an error occurrs. Importantly, the message is associated with the project that the user is currently logged in to. Finally, the new message is returned. Any

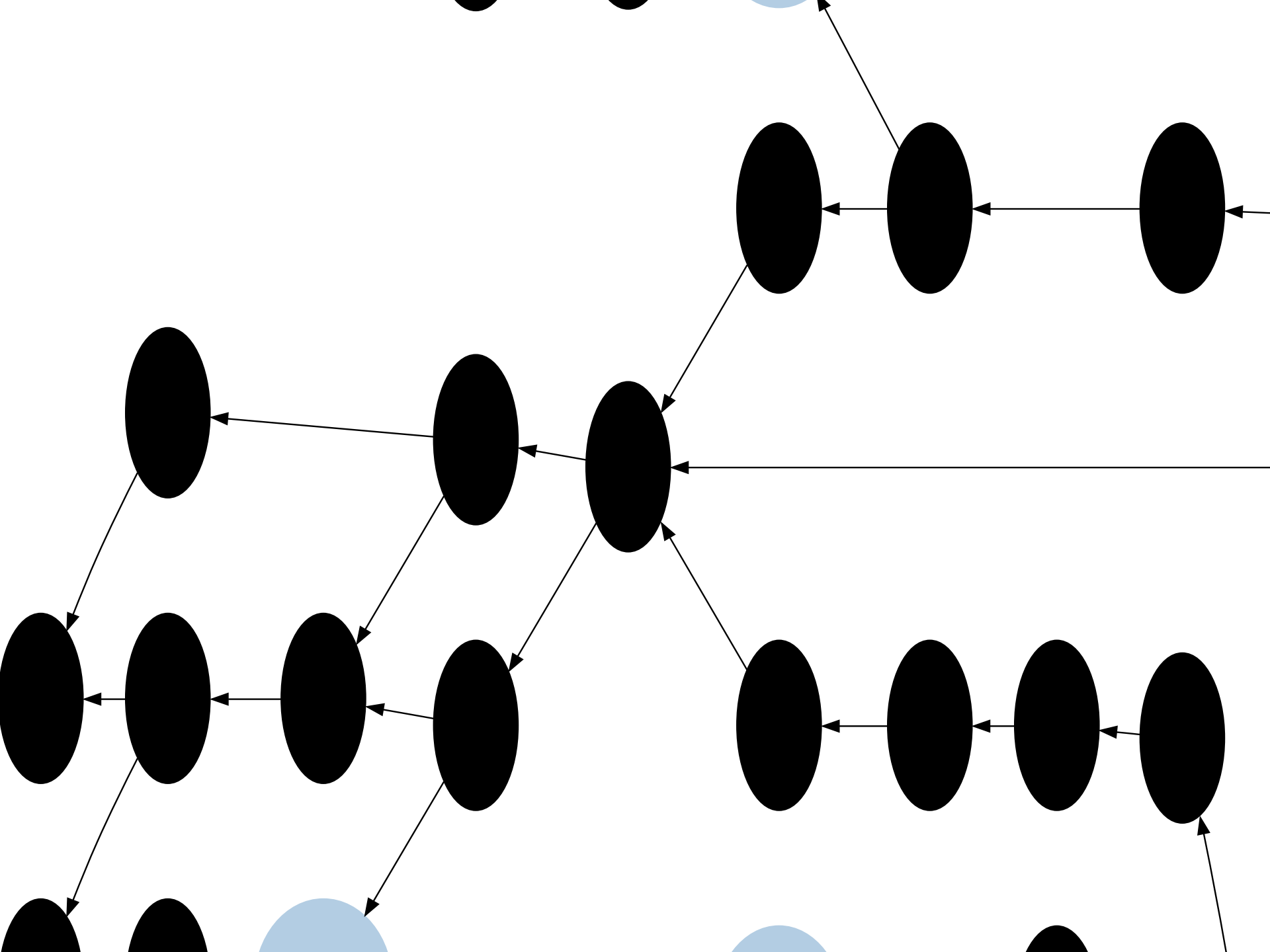
Future Goals

- More testing, especially of the GUI
- HTTPS support, including certificates
- Conflict resolution
- Dashboard screen
- Filters on defect search tab

Results

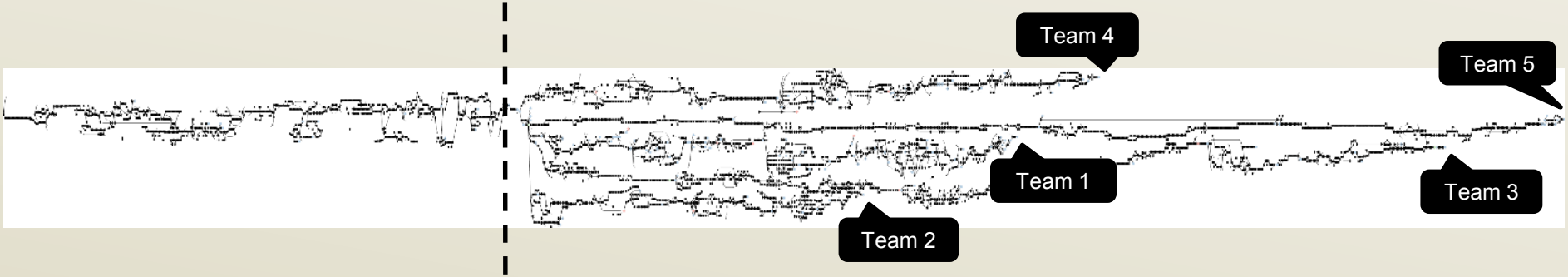
- Studied D Term Software Engineering
  - Teams developed a Requirements Management module
  - Three MQP team members were coaches





761 commits
16,705 Physical SLOC

3,287 more commits
42,039 more SLOC

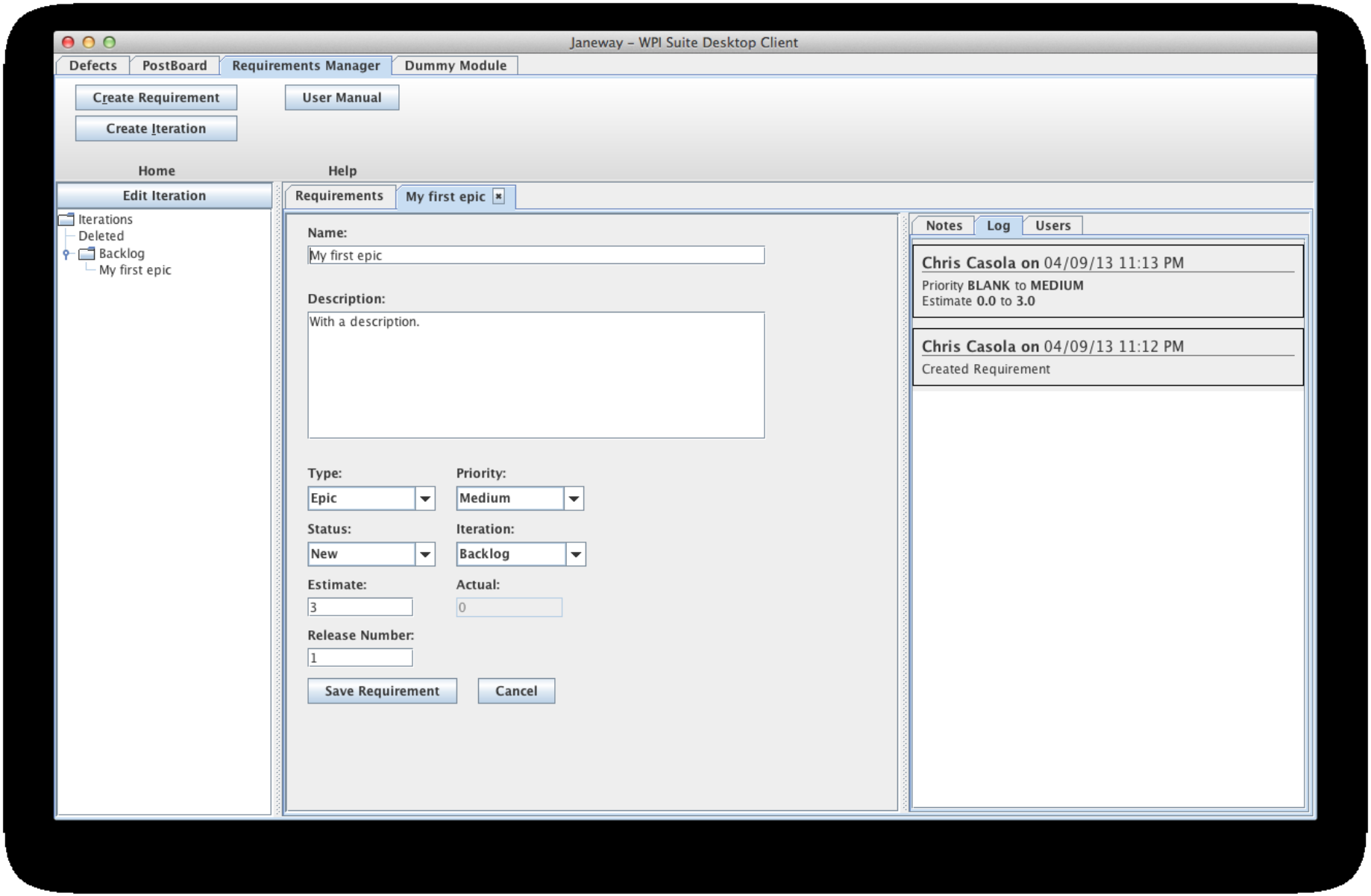


A Term

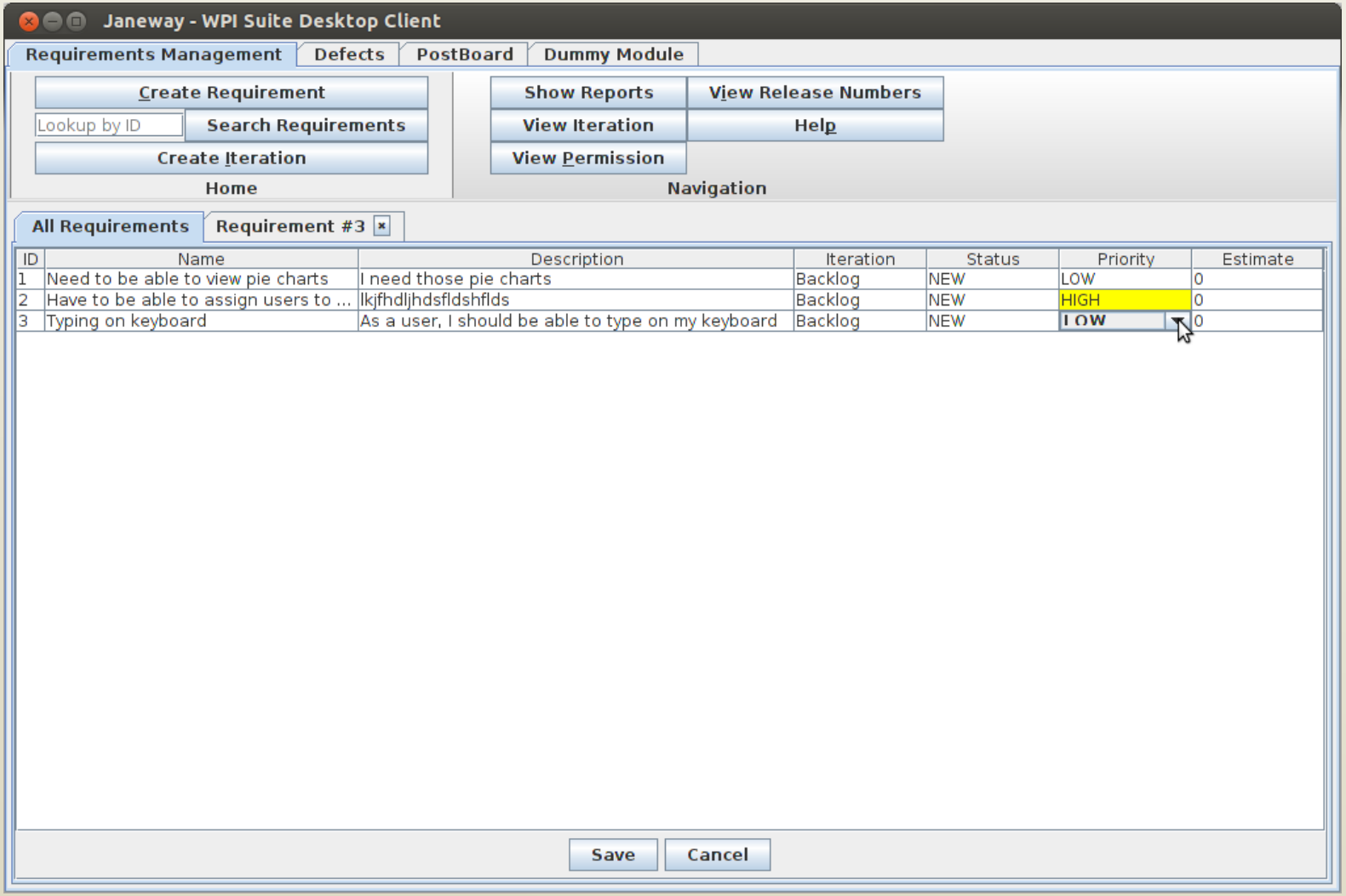
D Term

5 weeks later

D Term Software Engineering



D Term Software Engineering



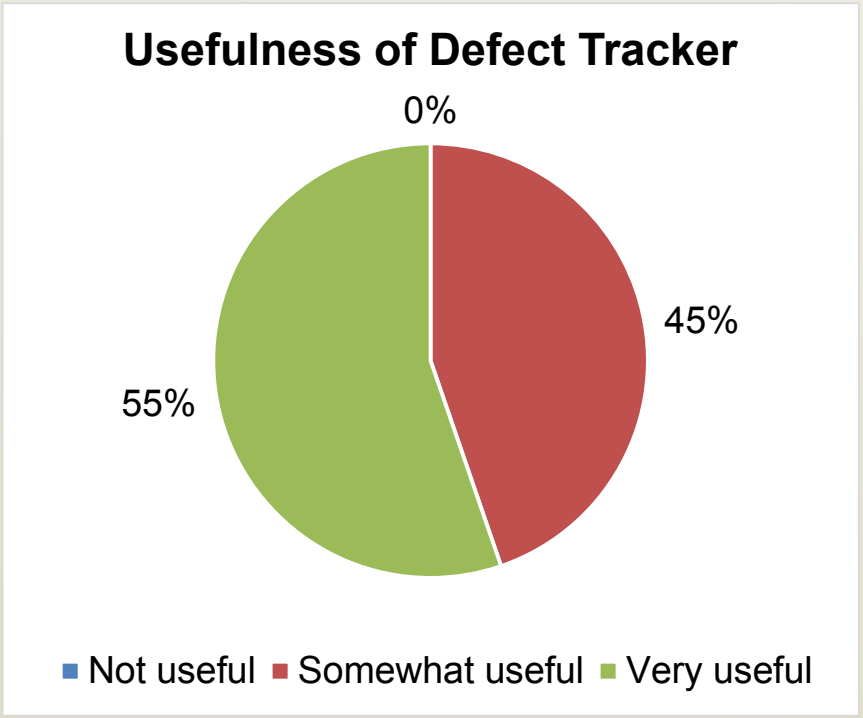
Project requirements

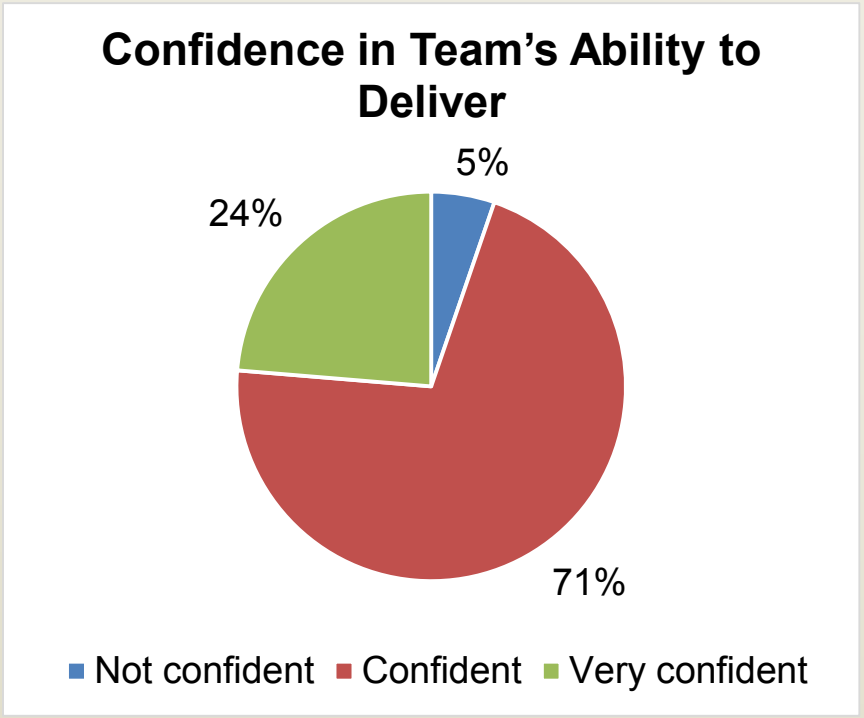
id	Name	Description	Iteration	Status	Priority	Estimate
1	qertytrh	gfhgfhgfhgdfwe	NONE	NEW	LOW	0
2	hjklqw	hjklhwe	NONE	OPEN	NONE	650
3	gh	This is a well thought out description. It holds text. Make sure to do this this this that hahahahaha	4wthgdr4	IN_PROGRESS	NONE	5487
4	safg	sdfgsd	4wthgdr4	IN_PROGRESS	NONE	454
5	h	ui	NONE	NEW	NONE	0
6	fg	ng	NONE	NEW	NONE	0

Requirement - fg

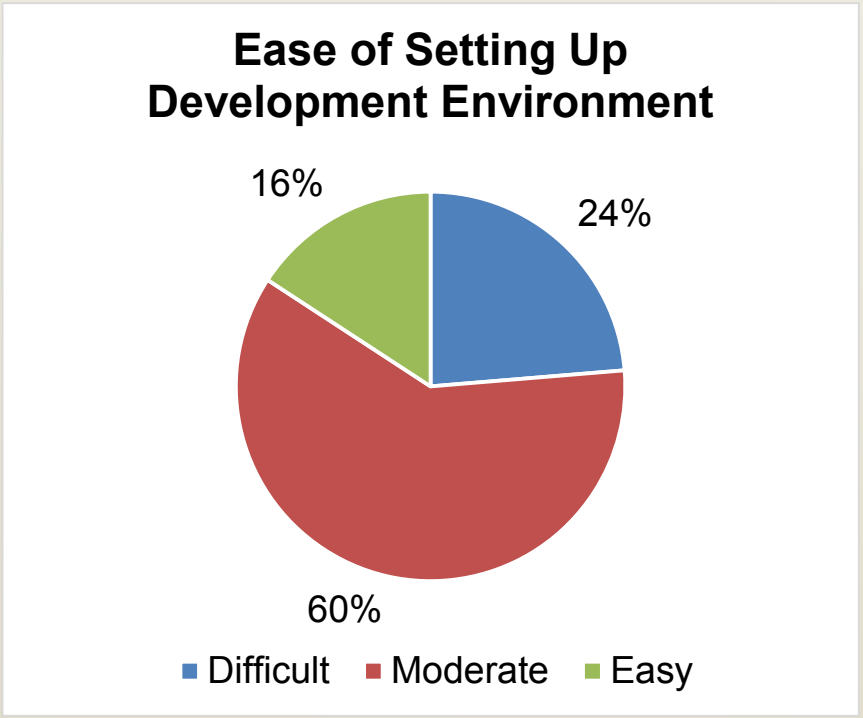
ID: 6
Name: fg
Description:
ng
Iteration: NONE
Status: NEW
Priority: NONE
Estimate: 0
Actual effort: 0
Date of creation: Apr 12, 2013 3:27:41 PM
Date of last modification: Apr 12, 2013 3:27:41 PM
Creator name: josh
Notes:

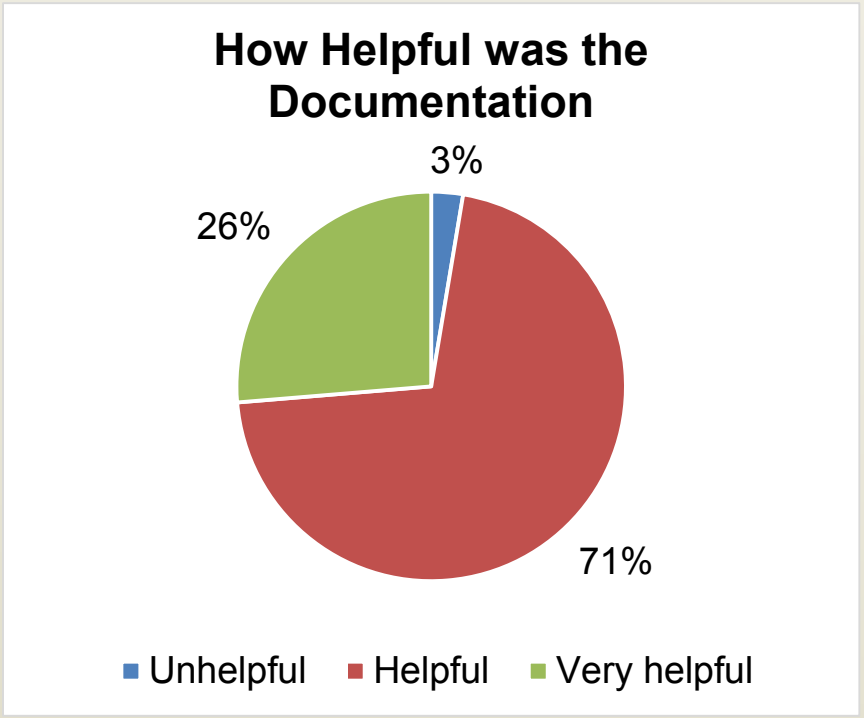
Survey Results





Survey Results





Overall Future Goals

- Virtual machine for development, and/or a bootstrap script
- Pagination
- Database migration
- Email support
- Consistent code style and CodePro audit rules

Open Source Community

Repo: github.com/fracture91/wpi-suite-tng

- Governance policy established
- Alumni have expressed interest in contributing
- Will be presented to education community, other schools
- Pull requests welcome!

github.com/fracture91/wpi-suite-tng

Questions?