shnip.it

Who? F. Hu J. Kay F. Johnson R. Saunders
J. Warburton

When? 2014-04-10

Project Infrastructure

Planning

Redmine

- Well-known, established Free project-management software
- Very customizable
- VCS integration
- . . . but kind of a pain to set up!

Backlogs

- Agile support for Redmine
- Group tasks into sprints
- Nice burndown charts
- Only thing we found that works with our Redmine version ☺

Facebook

- Good for informal conversations

Planning

Redmine

- Well-known, established Free project-management software
- Very customizable
- VCS integration
- . . . but kind of a pain to set up!

Backlogs

- Agile support for Redmine
- Group tasks into sprints
- Nice burndown charts
- Only thing we found that works with our Redmine version ☺

Facebook

- Good for informal conversations

Planning

Redmine

- Well-known, established Free project-management software
- Very customizable
- VCS integration
- . . . but kind of a pain to set up!

Backlogs

- Agile support for Redmine
- Group tasks into sprints
- Nice burndown charts
- Only thing we found that works with our Redmine version ☺

Facebook

- Good for informal conversations

Collaboration

git

- Lets us work without stepping on each others' toes
- Keeps track of work put into the project via the commit log
- Write-up using TEX; can be put in git too!
- Mature, stable infrastructure. . .

GitHub

- Service for maintaining a 'canonical' centralized repository
- Also provides communication services like an issue tracker or code comments (but we didn't use these)

Collaboration

git

- Lets us work without stepping on each others' toes
- Keeps track of work put into the project via the commit log
- Write-up using TEX; can be put in git too!
- Mature, stable infrastructure. . .

GitHub

- Service for maintaining a 'canonical' centralized repository
- Also provides communication services like an issue tracker or code comments (but we didn't use these)

Concept

Domain

- **Snippets** are pieces of code in a particular language that somebody might want to insert into their program.
- **Boards** are organizational objects used to group snippets into semantically related sets, and apply permissions (to allow the system to be used with sensitive codebases).

Domain

- **Snippets** are pieces of code in a particular language that somebody might want to insert into their program.
- **Boards** are organizational objects used to group snippets into semantically related sets, and apply permissions (to allow the system to be used with sensitive codebases).

Requirements

Market Research

Features

- Code-oriented features like syntax highlighting and line numbering
- Ease of sharing
- Plausibility of editor integration

Competitors

- **GitHub Gists**, as well as pastebin sites like **pastebin.com**, are code-oriented and can be shared, but provide no API for editor integration.
- **Google Docs** can be shared, but is not designed for code, and has no friendly API for integration.
- **yasnippet** and similar snippet tools are integrated with the editor (often only one editor, and can't be reused between editors), but provide no means of sharing.

Market Research

Features

- Code-oriented features like syntax highlighting and line numbering
- Ease of sharing
- Plausibility of editor integration

Competitors

- **GitHub Gists**, as well as pastebin sites like **pastebin.com**, are code-oriented and can be shared, but provide no API for editor integration.
- **Google Docs** can be shared, but is not designed for code, and has no friendly API for integration.
- **yasnippet** and similar snippet tools are integrated with the editor (often only one editor, and can't be reused between editors), but provide no means of sharing.

Market Research

Features

- Code-oriented features like syntax highlighting and line numbering
- Ease of sharing
- Plausibility of editor integration

Competitors

- **GitHub Gists**, as well as pastebin sites like **pastebin.com**, are code-oriented and can be shared, but provide no API for editor integration.
- **Google Docs** can be shared, but is not designed for code, and has no friendly API for integration.
- **yasnippet** and similar snippet tools are integrated with the editor (often only one editor, and can't be reused between editors), but provide no means of sharing.

Market Research

Features

- Code-oriented features like syntax highlighting and line numbering
- Ease of sharing
- Plausibility of editor integration

Competitors

- **GitHub Gists**, as well as pastebin sites like **pastebin.com**, are code-oriented and can be shared, but provide no API for editor integration.
- **Google Docs** can be shared, but is not designed for code, and has no friendly API for integration.
- **yasnippet** and similar snippet tools are integrated with the editor (often only one editor, and can't be reused between editors), but provide no means of sharing.

Functional Requirements

Research

To get an idea of what users would want from such a system, if at all, we produced a **questionnaire**.

Formulation

Since we used Scrum, we found it easiest to specify our functional requirements directly as **user stories**. This also helped us verify our functional tests.

Functional Requirements

Research To get an idea of what users would want from such a system, if at all, we produced a **questionnaire**.

Formulation Since we used Scrum, we found it easiest to specify our functional requirements directly as **user stories**. This also helped us verify our functional tests.

Non-Functional Requirements

These were imposed on us by

- The university, the team, and the nature of the task
- The possibility that we might store sensitive data

Non-Functional Requirements

These were imposed on us by

- The university, the team, and the nature of the task
- The possibility that we might store sensitive data

Implementation

Front-End

- We decided to create a Web-based user interface using **HTML5/CSS3/JavaScript** for ease of deployment.
- **jQuery** was used for some special effects.
- **Backbone.js** for communicating with the server when loading or storing data dynamically.

Front-End

- We decided to create a Web-based user interface using **HTML5/CSS3/JavaScript** for ease of deployment.
- **jQuery** was used for some special effects.
- **Backbone.js** for communicating with the server when loading or storing data dynamically.

Front-End

- We decided to create a Web-based user interface using **HTML5/CSS3/JavaScript** for ease of deployment.
- **jQuery** was used for some special effects.
- **Backbone.js** for communicating with the server when loading or storing data dynamically.

Back-End

- **Python** since it is already familiar to everyone in our group and has some good tools for rapid Web development.
- **Django**, a well-respected MVC Web framework, provided an assortment of useful development tools.
- **PostgreSQL** is the most standards-compliant Free database we found.
- But we also used **SQLite** for distributed development — Django makes the choice of SQL server mostly transparent.

Back-End

- **Python** since it is already familiar to everyone in our group and has some good tools for rapid Web development.
- **Django**, a well-respected MVC Web framework, provided an assortment of useful development tools.
- **PostgreSQL** is the most standards-compliant Free database we found.
- But we also used **SQLite** for distributed development — Django makes the choice of SQL server mostly transparent.

Back-End

- **Python** since it is already familiar to everyone in our group and has some good tools for rapid Web development.
- **Django**, a well-respected MVC Web framework, provided an assortment of useful development tools.
- **PostgreSQL** is the most standards-compliant Free database we found.
- But we also used **SQLite** for distributed development — Django makes the choice of SQL server mostly transparent.

Back-End

- **Python** since it is already familiar to everyone in our group and has some good tools for rapid Web development.
- **Django**, a well-respected MVC Web framework, provided an assortment of useful development tools.
- **PostgreSQL** is the most standards-compliant Free database we found.
- But we also used **SQLite** for distributed development — Django makes the choice of SQL server mostly transparent.

Development Model

Development Model

Scrum

- Divide project up into flexibly-scheduled sprints — suits university schedule (exams &c.)!
- Respond rapidly to changing requirements as we discovered them

TDD

- Constantly check our code for errors
- Particularly important in Python, as we have no type system to guarantee invariants

Development Model

Scrum

- Divide project up into flexibly-scheduled sprints — suits university schedule (exams &c.)!
- Respond rapidly to changing requirements as we discovered them

TDD

- Constantly check our code for errors
- Particularly important in Python, as we have no type system to guarantee invariants

Design

Video

Testing

Functional Testing

Formulation

- Derived directly from functional requirements
- Powerful means of validation!

Implementation

- **Django** provides a framework for running tests against a live server instance
- Used the **Selenium** Web driver with the **PhantomJS** headless browser to test through the frontend exactly as the user would see it

Functional Testing

Formulation

- Derived directly from functional requirements
- Powerful means of validation!

Implementation

- **Django** provides a framework for running tests against a live server instance
- Used the **Selenium** Web driver with the **PhantomJS** headless browser to test through the frontend exactly as the user would see it

Unit Testing

- **Python** provides a unit testing framework.
- **Django**'s MVC architecture reduces coupling and makes it possible to test individual components separately from the rest of the system.

Evaluation

Conclusion

Questions?