Integrating Version Control into AgiPlanner
Plan:

- Motivations
- Communication
- Technology Stack
- Tracking Commits
- Future Development
- Questions
What's Agile?

- Group of software dev methods
- Iterative and incremental dev
- Requirements and solutions evolve
- Collaboration between self-organizing, cross-functional teams

- Wikipedia
Tools:

- Standup / SCRUM
- Sprints
- Kanban
What's AgiPlanner?
Others:
Why Version Control?

- Track history
- Organize collaboration

- Git
- SVN
- Perforce
- TFS
Communication

Github
Webhooks

AgiPlanner
Web Services
Webhooks
Setup

Webhooks / Add new

We’ll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you’d like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *
https://example.com/postreceive

Payload version
application/vnd.github.v3+form

Which events would you like to trigger this webhook?
- Just the push event.
- Send me everything.
- Let me select individual events.

- Push
  Git push to the repository.

- Delete
  Branch or tag deleted.

- Deployment
  Repository deployed.

- Pull Request
  Pull Request opened, closed, or synchronized.

- Commit comment
  Commit or diff commented on.

- Issue comment
  Issue commented on.

- Fork
  Repository forked.

- Member
  Collaborator added to a non-organization repository.

- Release
  Release published in the repository.

- Create
  Branch, or tag created.

- Status
  Commit status updated from the API.

- Deployment status
  Deployment status updated from the API.

- Pull Request review comment
  Pull Request diff commented on.

- Issues
  Issue opened or closed.

- Watch
  User watches the repository.

- Public
  Repository changes from private to public.

- Team add
  Team added or modified on a repository.

- Gollum
  Wiki page updated.

- Active
  We will deliver event details when this hook is triggered.

Add webhook
We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you’d like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *
https://example.com/postreceive

Payload version
application/vnd.github.v3+form

Which events would you like to trigger this webhook?
- Just the push event.
- Send me everything.
- Let me select individual events.

Push
Git push to the repository.

Create
Branch, or tag created.

Status
Which events would you like to trigger this webhook?

- Just the push event.
- Send me everything.
- Let me select individual events.

- **Push**
  Git push to the repository.

- **Delete**
  Branch or tag deleted.

- **Deployment**
  Repository deployed.

- **Pull Request**
  Pull Request opened, closed, or synchronized.

- **Commit comment**
  Commit or diff commented on.

- **Issue comment**
  Issue commented on.

- **Fork**
  Repository forked.

- **Member**
  Collaborator added to a non-organization repository.

- **Release**
  Release published in the repository.

- **Create**
  Branch, or tag created.

- **Status**
  Commit status updated from the API.

- **Deployment status**
  Deployment status updated from the API.

- **Pull Request review comment**
  Pull Request diff commented on.

- **Issues**
  Issue opened or closed.

- **Watch**
  User watches the repository.

- **Public**
  Repository changes from private to public.

- **Team add**
  Team added or modified on a repository.

- **Gollum**
  Wiki page updated.

---

- **Active**
  We will deliver event details when this hook is triggered.

**Add webhook**
The Post

"commits": [
  {
    "id": "1075773a33ca8b06ba4478eeb71cf71e98c00a17",
    "distinct": true,
    "message": "Test update of readme. [card:290]",
    "timestamp": "2014-02-26T23:51:15-08:00",
    "url": "https://github.com/kyle-rader/Web/commit/1075773a33ca8b06ba4478eeb71cf71e98c00a17",
    "author": {
      "name": "kyle-rader",
      "email": "kyle.rader@ieee.org",
      "username": "kyle-rader"
    },
    "committer": {
      "name": "kyle-rader",
      "email": "kyle.rader@ieee.org",
      "username": "kyle-rader"
    },
    "added": [
    ],
    "removed": [
    ],
    "modified": [
      "README.md"
    ]
  }
],
The Webservice

http://kyle.groupsrum.stevestedman.com/
VCShook/git/7077914e75e9706a5d7103396846017c

Webservice: VCShook

Repository code:
7077914e75e9706a5d7103396846017c

Version control system: git
Technology Stack

- Web Server: Apache
- Web Framework: non-MVC PHP w/ Foundation
- Database: MySQL
- VCS: Git via Github

**How I develop (at the moment):**

- Cygwin
- Notepad++ w/ SFTP extension
- Chrome
- MySQL Workbench
Tracking Commits

Important commit info (from webhook payload):

- Commit hash
- Author

Other important data:

- Repository
- Repository Owner (github account)
Repo 'Registration'

Provides a layer of security
Repo 'Registration'

Administration: VCS Integration

<table>
<thead>
<tr>
<th>Git</th>
<th>SVN</th>
<th>TFS</th>
<th>CVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Github Account Name</td>
<td>Repository</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Registered Repositories

<table>
<thead>
<tr>
<th>Github Account</th>
<th>Repository</th>
<th>Webhook URL</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>kyle-rader</td>
<td>Web</td>
<td><a href="http://kyle.groupscrum.stevestedman.com/VCShook/git/7077914e75e9706a5d7103396846017c">http://kyle.groupscrum.stevestedman.com/VCShook/git/7077914e75e9706a5d7103396846017c</a></td>
<td>Delete</td>
</tr>
</tbody>
</table>

Provides a layer of security
Getting Data to Display

- Getting commits (to the white board!)
- (this is still in development)
Visual Display (to-do)

Add a tab/page to user stories
Visual Display (to-do)

Add an ajax web service call to return the commits for this story
Summary

- User registers a repo
- Add webhook url to that repo
- Pushes/ pull requests post our web service
- We parse for "[card: ##, ## ...]" syntax
- Log commits and correlate to user stories
Future Development

- Github-services
- using full Github API (manage Github)
- D3 data driven JavaScript library
Questions?