ROW5 Sprint 4 Review

ROW Team 5

Amsterdam University of Applied Sciences
https://rescueonwheels.github.io/

Januari 14, 2019

Introduction

Christiaan van Arum	Developer
Raphaël Bunck	Scrum Master
Nino van Galen	Developer
Martijn Vegter	Product Owner

Overview

1 Introduction

2 Operational scenario

3 Sensors and actuators

4 Communication

5 User Interface

6 Documentation

7 Innovatation

8 Scrum

9 Quality assurance

10 End

Operational scenario

- Building collapse due to:
  - Earthquake;
  - Explosion.
- Natural disasters:
  - Avalanche;
  - Landslide.
- General exploration

Sensors and actuators

Sensor(s):

- Distance sensor (ultrasonic sensor HC-SR04).

Actuator(s):

- Custom double axis servo platform.

Other:

- Fisheye lens.

Communication

Rover ↔ Cockpit:

- Socket.IO.

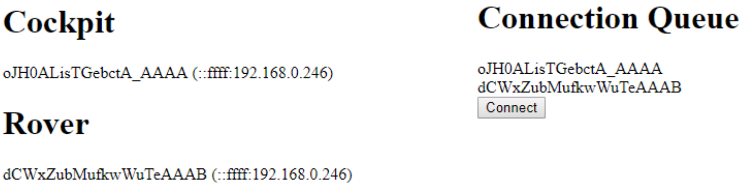
Rover → Tincidunt:

- H.264 over HTTP.

Cockpit ← Controller:

- Bluetooth;
- USB.

User Interface: Epicenter



Cockpit

oJH0ALisTGebctA_AAAA (::ffff:192.168.0.246)

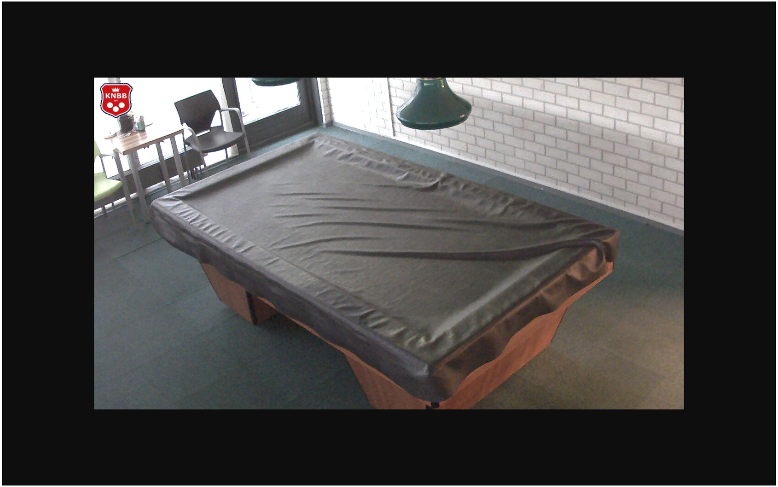
Rover

dCWxZubMufkwWuTeAAAB (::ffff:192.168.0.246)

Connection Queue

oJH0ALisTGebctA_AAAA
dCWxZubMufkwWuTeAAAB

User Interface: Chrome



User Interface: Tincidunt

Documentation

- Rover Rescue System - Business Case
- Rover Rescue System - (Technical) Documentation
- Rover Rescue System - Manual - Application
- Rover Rescue System - Manual - Epicenter
- Rover Rescue System - Manual - Rover
- Rover Rescue System - Project File
- Rover Rescue System - Sprint Review 1
- Rover Rescue System - Sprint Review 2
- Rover Rescue System - Sprint Review 3
- Rover Rescue System - Sprint Review 4

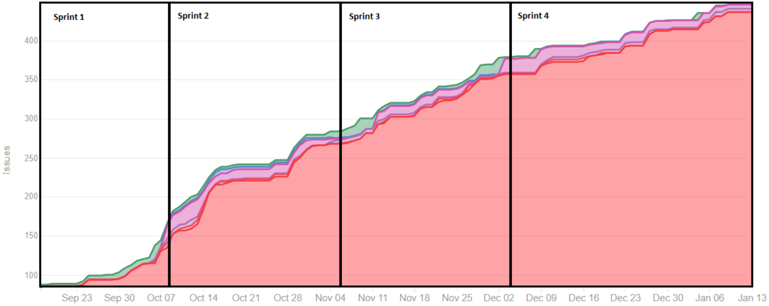
Innovatation

- Virtual Reality as video output;
- Virtual Reality connected to the camera;
- Automated prevention systems:
    - Auto-stop to prevent crashes;
    - Auto-stop based on communication events;
    - Auto-reset of camera view based on communication events;
    - Scalable for large scale operations.

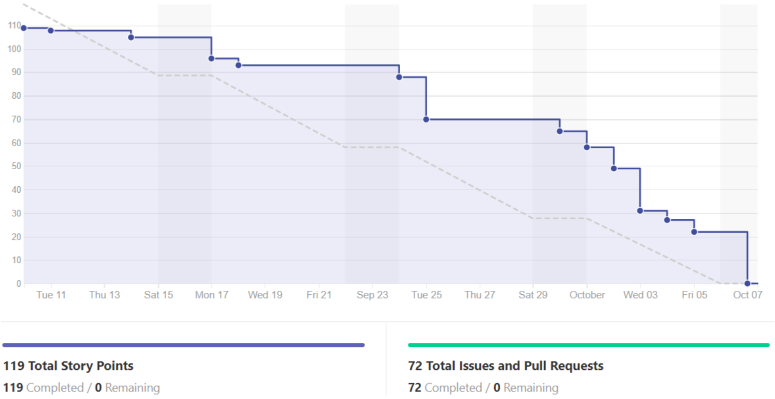
Scrum

- Use of GitHub projects;
- Use of GitHub because of third-party integrations;
- Use of ZenHub for automated issue tracking;
  - Isssues
  - Epics
  - Pull Requests
- Use of ZenHub for Burndown and Velocity tracking;

Scrum: Cumulative Flow



Scrum: Burndown sprint 1



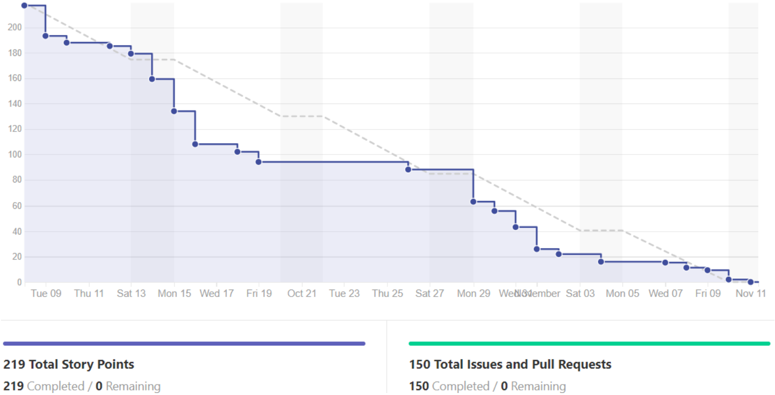
119 Total Story Points

119 Completed / **0** Remaining

72 Total Issues and Pull Requests

72 Completed / **0** Remaining

Scrum: Burndown sprint 2



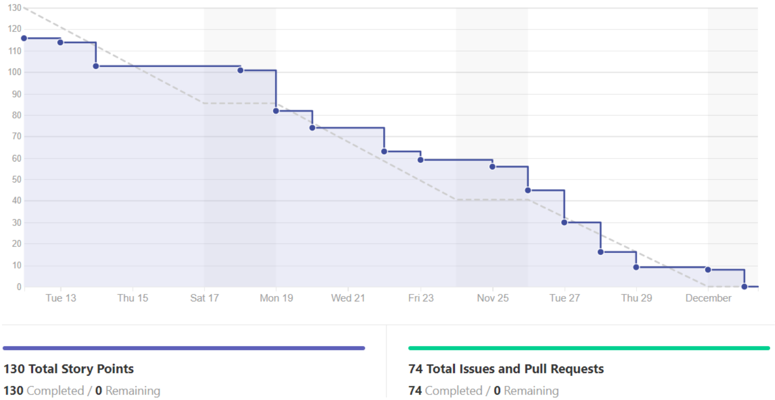
219 Total Story Points

219 Completed / **0** Remaining

150 Total Issues and Pull Requests

150 Completed / **0** Remaining

Scrum: Burndown sprint 3



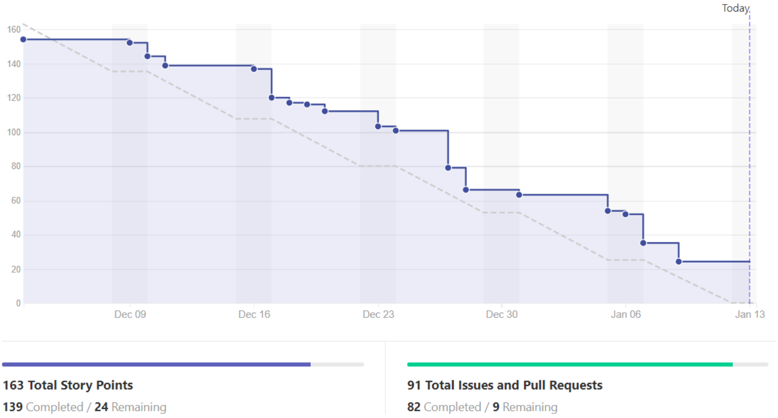
130 Total Story Points

130 Completed / **0** Remaining

74 Total Issues and Pull Requests

74 Completed / **0** Remaining

Scrum: Burndown sprint 4



163 Total Story Points

139 Completed / **24** Remaining

91 Total Issues and Pull Requests

82 Completed / **9** Remaining

Quality assurance

- Use of GIT submodules;
- Custom mocks for simulation usage;
- Protected branches with following rules:
  - Require pull request reviews before merging;
  - Require status checks to pass before merging
    - Travis-CI used for tests and code style;
    - CodeClimate used for unbiased code quality;
    - Coveralls is used for code coverage.
- Definition of Done;
- Definition of Ready.

Any Questions?

The End