SAND2011-3876C

Using Agile at Sandia

Lessons Learned

June 15th, 2011

Hope Niblick
Web Applications Team Lead

Why go Agile?

- Change from a platform that did not easily support group developer efforts (Domino) to a highly collaborative one (Ruby on Rails)
- Expectation of group – to develop innovative software in very short time cycles
- Track record of cutting-edge ideas and initiatives
- Increased focus on project management and financial reporting – life as a service center

Sandia National Laboratories

Retrospect this

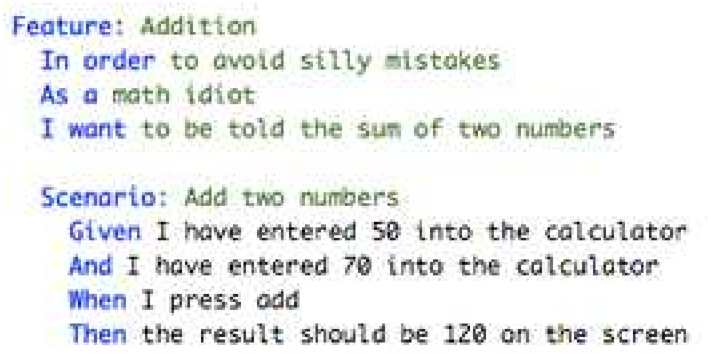
- Agile methodology includes the practice of retrospection, so here's ours
- What went well
- What didn't go well
- What we should do next time
- What we should never do again

What went well?

- Retrospectives
  - Let everyone express their opinions on the process
  - Gave a little breathing room to think things thru
  - Provided a baseline for accountability – if we said we would never do it again, it's now in writing
  - Fine tuning things that didn't quite go as well as we would have liked
- Scrum
  - Even team members who initially resisted came to like them

What went well? (con't)

- Tee shirt sizes instead of Fibonacci numbers for planning poker exercises
- Behavior Driven Development
  - Not always possible
  - When possible, helped to ensure that the developers were closely following customer's directions

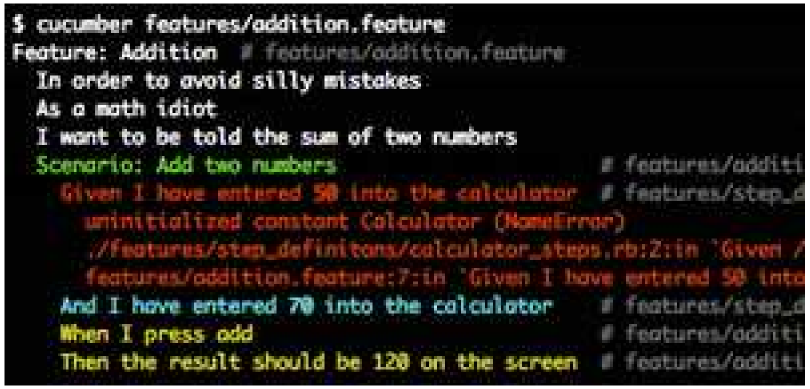


```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

  Scenario: Add two numbers
    Given I have entered 50 into the calculator
    And I have entered 70 into the calculator
    When I press add
    Then the result should be 120 on the screen
```

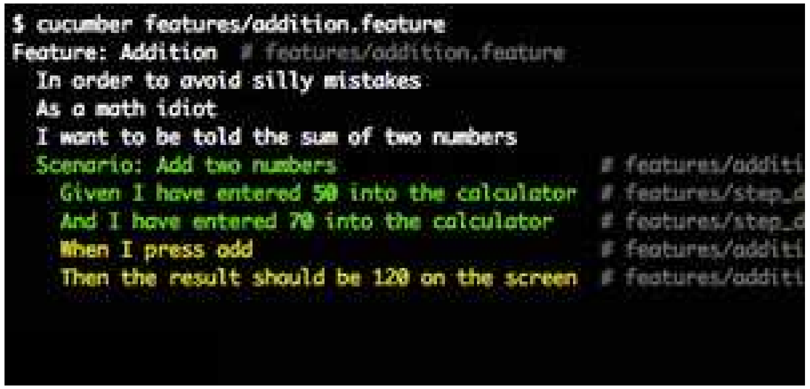
More BDD

```
Given /I have entered (.*) into the calculator/ do |n|
  calculator = Calculator.new
  calculator.push(n.to_i)
end
```



```
$ cucumber features/addition.feature
Feature: Addition  # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
  Scenario: Add two numbers                      # features/additi
    Given I have entered 50 into the calculator  # features/step_d
      uninitialized constant Calculator (NameError)
      ./features/step_definitons/calculator_steps.rb:2:in `Given /
      features/addition.feature:7:in `Given I have entered 50 into
    And I have entered 70 into the calculator    # features/step_d
    When I press add                             # features/additi
    Then the result should be 120 on the screen  # features/additi
```

```
class Calculator
  def push(n)
    @args ||= []
    @args << n
  end
end
```



```
$ cucumber features/addition.feature
Feature: Addition  # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
  Scenario: Add two numbers                      # features/additi
    Given I have entered 50 into the calculator  # features/step_d
    And I have entered 70 into the calculator    # features/step_d
    When I press add                             # features/additi
    Then the result should be 120 on the screen  # features/additi
```

What didn’t go so well?

- Some software not up to the task
  - Slowed down scrums
  - Wasted time
  - Provided little benefit
- Scrums that turned into status updates
  - Eating into short amount of meeting time
  - Wasting some team members’ time
- Resistance from customers
- Resistance from team members (not always conscious)
- Fiscal year strictures
- Not allowing enough time for meetings

Sandia National Laboratories

What we’ll never do again

- Do an Agile project without the customer’s 100% buy-in on the process
- Use software that slows down/hinders scrums
- Let the team get disturbed in the middle of a sprint
- Allow outside forces pull us towards waterfall processes when we’re doing Agile – you can’t have all of both

Sandia National Laboratories

What we'll do next time

- All the things that worked well
- Explore new software that may help us
  - Time keeping software that should give team members a chance to advise on progress (@Task)
  - New version of TeamForge
- Better process for setting customer expectations and understandings
- Use more post-it notes
- Experiment with other time-management techniques (maybe Pomodoro)

Sandia National Laboratories

Questions?

- War stories you'd like to share?

Hope Niblick

Sandia National Lab / Livermore, CA

hniblic@sandia.gov