Week 4: Intro to Agile SDM and Rails

409232 Software Development Methods

Jim Buchan

Software Engineering Research Laboratory

- Clarity on Assignments 2
- Overview of Agile and tool support

Assignment 2

- Sprint 0 - weeks 3,4,5
- Sprint 1 weeks 6,7
- Sprint 2 weeks 8,9
- Sprint 3 weeks 10,11
- Delivery week 12
- 6 Teams of 6-8 members
  - Analyst x 1, testers x 2, coders x 2, scrum master x 1
  - 2 competing streams of 3 teams developing different versions of the app.
    - 1 team - input of information
    - 1 team - querying information
    - 1 team - workflow and administration

- Aim
- Workflow
- Getting information in to the repository
- External users using the repository
  - view/browse
  - rate
  - search
- Administering the repository

- Teams of 5-6
- Multi functional
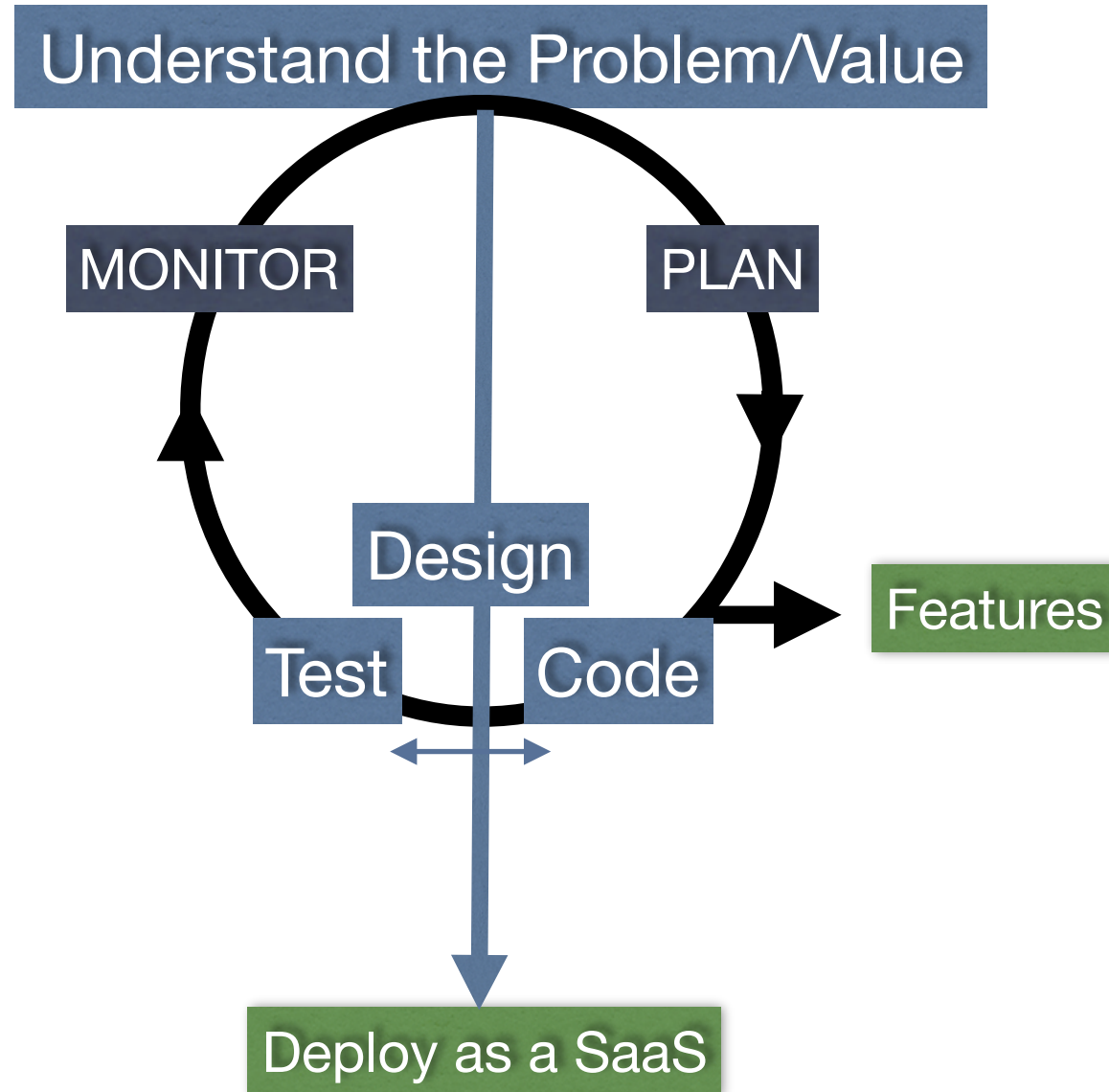 - analyst
  - testers
  - developers
  - scrum master

Organizational

1 Put the customer at the center.

2 Let the team self-organize.

3 Work at a sustainable pace.

4 Develop minimal software:

    4.1   Produce minimal functionality.

    4.2   Produce only the product requested.

    4.3   Develop only code and tests.

5 Accept change.

Technical

6 Develop iteratively:

    6.1   Produce frequent working iterations.

    6.2   Freeze requirements during iterations.

7 Treat tests as a key resource:

    7.1   Do not start any new development until all tests pass.

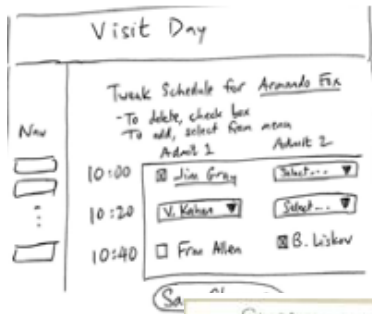   7.2   Test first.

8 Express requirements through scenarios.

Collaboratively developing an Application

Understand the Problem/Value

User Stories

Scenarios

Mockups

Personas

Design

Test ⟷ Code

Features

Behaviour Driven Development

Test First Design

Share Code Version Control

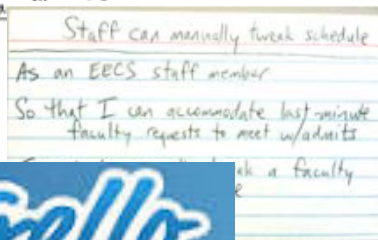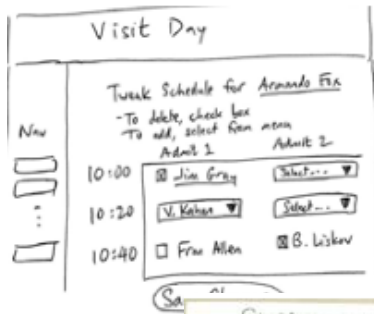Continuous Integration

Deploy as a SaaS

AUT | COMPUTER + MATHEMATICAL SCIENCES

SERL
Software Engineering Research Laboratory

Collaboratively developing an Application

Understand the Problem/Value

User Stories

Scenarios

Mockups — balsamiq®

Personas

Design   Features

Test   Code

Behaviour Driven Development — Cucumber

Test First Design — RSpec / Debugging Sucks! Testing Rocks!

Share Code Version Control — GitHub

Continuous Integration — Travis CI

heroku   **Deploy as a SaaS**   SERL

Software Engineering Research Laboratory

- Use of Rails, BDD
- Use of RSpec for automated testing
- Use of Travis CI for continuous integration
- Use of Cucumber for acceptance testing
- Use of Bootstrap, Font awesome for look and feel
- Use of GitHub for version control and code sharing
- Use of Heroku for cloud deployment
- Get product backlog - manage in Trello
- Sprint planning, release planning

Agile organisational Methods

- Teams/individuals share code within each stream

- Standup meeting every class

- Sprint planning before eac~~~~ and estimation)

- Pair pro~~~~ week

- ~~~~ end of each sprint

- Te~~~~ keep in touch f2f and electronically on a DAILY basis

"How evidence all of this for academic credit?
How document individual contribution?"

- This weeks practical exercise involves understanding the use of the model and controllers to modify the database.

- http://wpm4240new.herokuapp.com/movies

-

- Read Chapters 4 and 10 of the textbook.
  - This will be the topic of the quiz next week.
  - **Marks will be collected for this quiz.**
- Look up information on Behaviour Driven Development and Cucumber so you have an idea of the claimed benefits and how cucumber supports BDD.