Agile Methods

CS 3250
Software Testing

[Ammann and Offutt, “Introduction to Software Testing,” Ch. 4]
Agile Methods

• Agile methods recognize that
  • Software engineers are not good at developing requirements
  • Software projects change continuously
  • We do not anticipate many changes
  • Many of the changes we do anticipate are not needed
  • Requirements (and other non-executable artifacts) tend to go out of date very quickly

• Agile methods start small – with some behaviors and specific tests – and then evolve over time
Test Harness as Guardian

• What is correctness?
  • Traditional: universal
  • Agile: existential

• Limit view of correctness
  • Traditional: define all correct behavior completely at the beginning
  • Agile: define correctness of some behavior with specific tests
    • If the software behaves correctly on the tests, it is correct

Even as the software (including the test cases) evolve, the correctness of the system at any single point in time is subject to immediate verification by running the test set.
Test Harness Verify Correctness

- Tests must be automated
- Every test must include a test oracle (mechanism that can evaluate whether that test passes or fails)
- Tests (executable artifacts) replace the requirements (non-executable artifacts)
- Tests must be high quality and must run quickly
- Tests must be run every time changes are made to the software

Test harness runs all automated tests efficiently and reports results to the developers
Agile Airplane Testing

**Test harness:**
- Appearance matches
- Color coding in place
- Fly 6ft (or 2m) over table

**Implementation:**
- Fold airplanes

**Testing:**
- Verify with test harness
  (accept or reject)

[Image from http://www.twilightsoftwares.com]
Activity: Agile Airplane

Air shows have become popular spectator events. Air shows across the country have placed orders for planes, we need to fulfill them. Everyone here is part of the “Awesome Agile Aviation” Company and can coordinate in any way.

• Form a team of 8-10 members (1 – QA/tester; the rest – developers)

• Roles (each team):
  • QA/testers: Evaluate the product after each Sprint
  • Developers: Produce airplanes

• Economics: Each team has $40 in the bank. The company has a fixed cost (burn rate) of $12 per team per Sprint, and revenue from accepted orders

• Coverage criteria: “Done” criteria

• Test harness: appearance matches the picture; color coding in place; fly 6ft (or 2m) over table

• 3 Sprints. After each Sprint, QA/testers will evaluate the airplanes each team produces – “accept” or “reject”

[Adapted from Agile Airplane Game, by John Heintz, GistLabs]
Agile Airplane – Sprint 1

**Sprint 1 (5 minutes)**

- Developers:
  - Use the provided supplies, follow the production instruction#1 to produce airplanes
  - You have 1 minute to plan Sprint
  - You have 4 minutes to run Sprint

- QA/tester:
  - Suggest developers how to ensure the products meet the acceptance criteria
  - Cannot help producing the airplanes

**End of Sprint 1 (2-4 minutes)**

- Developers:
  - Demonstrate to QA/tester

- QA/tester:
  - If the airplanes pass the tests, accept the planes, group batches of “done” planes
  - If the airplanes fail the tests, tell the teams about acceptance criteria and reject the planes (“rip” them)
  - Pay $20 for every 15 “done” airplanes (i.e., add $20 to the team financial chart)
  - Update the team financial chart with $12 burn rate
Agile Airplane – Sprint 2

**Sprint 2 (5 minutes)**

- Developers:
  - Use the provided supplies, follow the production instruction#2 to produce airplanes
  - You have 1 minute to plan Sprint
  - You have 4 minutes to run Sprint

- QA/tester:
  - **Suggest** developers how to ensure the products meet the acceptance criteria
  - Cannot help producing the airplanes

**End of Sprint 2 (2-4 minutes)**

- Developers:
  - Demonstrate to QA/tester

- QA/tester:
  - If the airplanes **pass** the tests, **accept** the planes, group batches of “done” planes
  - If the airplanes **fail** the tests, tell the teams about **acceptance criteria** and **reject** the planes (“rip” them)
  - Pay $30 for every 10 “done” airplanes (i.e., add $30 to the team financial chart)
  - Update the team financial chart with $12 burn rate
Agile Airplane – Sprint 3

**Sprint 3 (5 minutes)**

- **Developers:**
  - Use the provided supplies, follow the production instruction#3 to produce airplanes
  - You have 1 minute to **plan** Sprint
  - You have 4 minutes to **run** Sprint

- **QA/tester:**
  - **Suggest** developers how to ensure the products meet the acceptance criteria
  - Cannot help producing the airplanes

**End of Sprint 3 (2-4 minutes)**

- **Developers:**
  - Demonstrate to QA/tester

- **QA/tester:**
  - If the airplanes **pass** the tests, **accept** the planes, group batches of “**done**” planes
  - If the airplanes **fail** the tests, tell the teams about **acceptance criteria** and **reject** the planes (“ripping” them)
  - Pay $40 for every 5 “**done**” airplanes (i.e., add $40 to the team financial chart)
  - Update the team financial chart with **$12 burn rate**
Activity: Wrap-up

• How were test harnesses used?
• How were criteria applied?
• How were acceptance tests applied?
• How likely had the quality been improved from Sprint to Sprint?