Company Background

- Canadian Pacific Railway is a very traditional company
  - 125 years since last spike driven on transcontinental rail line

- Embraces traditional waterfall or document-driven development practices
  - It works for laying track and building bridges!

- Usability testing not part of CPR culture

- Agile is relatively new to CPR I.T.
Project Background

- Used eXtreme Programming “out of the box”
  - 2 week iterations with IPM and Retrospective
  - Accepted Responsibility
  - Emergent design / refactoring
  - Continuous Integration (Cruise Control)
  - Unit Test-driven Development - mostly
  - Pair Programming - about 50% of the time
  - Onsite Customer
    - Different floor
    - Part time, revolving door
  - Storytest-Driven Development
    - Mostly using Fit for test automation
    - less than 50% of stories had Fit tests

Key Learnings

1. Focus on Usability Improved User Acceptance and Reduced Rework
   
   **Before:**
   - First release had many issues raised by users during acceptance testing
     - Many were usability-related issues
     - Features worked but were awkward
   - Resulted in 6 weeks of rework for entire development team
Key Learnings

*With Usability:*
- Introduced Paper Prototyping and Wizard of Oz testing for 2nd release.
- Engaged key end users for usability testing sessions.
  - End-users felt their opinions were valued.
- Allowed for 6 person-weeks of usability enhancements as part of product backlog.
- High end-user satisfaction with application
  - E.g. e-mail from ecstatic end-user
    “I love this application…”

2. Paper Prototyping Early In Release Cycle Helped Make Vision More Tangible to Entire Team

*Before:*
- In first release, team didn’t have a clear vision of what “done looks like”.
- Features “emerged” as the project unfolded and some important functionality was discovered very late (during testing.)
Key Learnings

With Paper Prototyping:
- Paper Prototype clearly showed what finished application might look like.
- Stimulated conversions/discussions about functionality and business processes.
- User Stories described behavior associated with UI elements.
- Paper Prototype provided context for each User Story.
- Release Backlog was visible – Story Cards associated with parts of Paper Prototype made it clear what wasn’t done.
- PP helped validate completeness of User Story list.

3. Paper Prototype / Wizard of Oz
   Testing was easy to do and saved effort
   - Paper, Glue, Scissors, Spreadsheet
   - Done by PM and Agile coach in spare time.
   - Done without any usability expert or experience.
   - Roughly 4 person days to build PP.
Paper Prototypers in Action!

Wizard of Oz Testing

- Roughly 1 team day to conduct Wizard of Oz tests:
  - 2 developers playing computer & coprocessor
  - 2-3 observers (developers + business)
  - Facilitator (business SME)

- Test Sessions:
  - 3 Test sessions of 1 hour each
  - Tested with pairs of end users (co-discovery)
  - Performed 3 tasks defined by business SME

- Did 2nd round of usability testing on early version of working software
Wizard of Oz Testing

Test Session Process:
- Describe task
  - “E-mail” from customer asking for a specific rate(s)
- Have users try to discover how to do without any coaching / help
- Computer responds to users’ actions
  - Customizes and lays down next screen
- Observers record usability issues/defects
  - Missing controls, unclear field/button labels
  - Impossible tasks
- Final debrief with all users/observers present
- Present final list/plans to address issues (later)
Key Learnings

4. Paper Prototype provided way of organizing Story Cards for Iteration Planning

Before:
- Had all “half-baked” story cards on Project Backlog wall
- Organized by arbitrary categorization
- Business or Development selected stories somewhat randomly
- Difficult to tell whether list was complete
- Business had difficulty prioritizing stories

With Paper Prototype “Story Board”:
- Put all the story cards onto PP near the UI elements they introduced
- Story Cards described the Behavior associated with elements (complementary)
- Early Iterations focused on happy path UI elements; stories associated with the Happy Path elements were easy to determine
- Later iterations delivered User Stories that added additional UI elements
- Easy to tell what parts of application were done (no story cards) and what remained (with story cards)
UI Story Board  (Late in the project)

UI Storyboard Detail
Key Learnings

5. Agile + Usability = Even Better Result

UxD:
- UxD often considered as conflicting with Agile methodology
  - Due to emphasis on Big Design Up Front

Agile:
- Issues around organizing User Stories without the context
  - Categorization is arbitrary
  - Hard to see “big picture” in sea of Story Cards
  - Hard to build self-consistent UI incrementally

With Agile + UxD:
- Found UxD and Agile to be complementary:
  - User Stories described behavior associated with UI elements
  - UI Provided context for each User Story
- Right-sizing & “right-timing” design”
  - “Just Enough” UI Design Up Front (UIDUF)
  - “Just in Time” (emergent) software design
- Reduced cost by eliminating software rework:
  - Paper Prototype is just as testable as software
  - Reworking PP much cheaper than reworking software
  - Very little usability enhancement required after acceptance testing
Summary

- Easy to do
- Low cost
- Complements Agile practices

JUST TRY IT!

References

- Book: Paper Prototyping
  - The Fast and Easy Way to Design and Refine User Interfaces
  - By: Carolyn Snyder
  - Published by: Morgan Kaufman 2003

  This is the book we wish we had known about when we started. It answers many of the questions we raised and had to answer for ourselves.