A Test-Driven Approach for Extracting Libraries of Reusable Components from Existing Applications

Paper by:
Elaf Selim, Yaser Ghanam, Chris Burns, Teddy Seyed, and Frank Maurer

Department of Computer Science, University of Calgary
Outline

- XP & Reuse
- Proposed Approach...
- Case Study & Extraction Steps
- Approach Benefits & Challenges
- Next Steps...
- Summary
XP & Reuse

- XP aims at
  - Developing products **quickly**
  - Satisfying **current** customers
  - Keeping it **Simple**

- **YAGNI** – **You Ain’t Gonna Need It**
  - Time not spent on designing components for future reuse
XP & Reuse

- **Reuse** has some benefits...
  - Saves development time of new software
  - Reduces maintenance costs. Ex: using libraries

- So, How can XP benefit from reuse without wasting time upfront?
Proposed Approach...

- XP
- Extractive
- Reusable Components
- Iterative
- Opportunistic Reuse
- On Demand
- Reactive
- Refactoring
- Incremental
- Test-driven
- Acceptance Tests
- User Stories
- Usage Examples
Library of Reusable Components

App # 1
- User Scenarios

App # 2
- User Scenarios

App # 3
- User Scenarios

Uses

Refactoring
Acceptance Tests Role

- Capture the user stories to represent the requirements of the Application

- Determine the components to be reused and the variability options to be added to them.

- A Safety Net to make sure refactoring:
  - maintains the behavior of the first App
  - fulfills the requirements of the new App.
Case Study: App 1 – eGrid

- A multi-touch tabletop application

- Designed for interacting with GIS maps in the Control Room of an Electricity Company.

- Developed using an XP approach, relying heavily on Customer Interactions.
Case Study: App 2 – eWell

- Designed for an Oil Exploration Firm
- To visualize sensor data for detecting oil concentrations.
- Data is captured and stored in GIS.

Questions:
- Which scenarios are shared by eGrid and eWell?
- Which components of eGrid should be separated into the API?
Extraction Steps

1. **Write acceptance tests for eWell**
   - Scenarios related to the map frame

2. **Compare the User Stories of eGrid and eWell**
   - To identify reuse opportunities and variability points
   - Ex: different gestures for creating the map frame
Extraction Steps

3. Analyze the Acceptance Tests of eGrid and eWell
   - To determine the implementation changes needed
   - Ex: different constructors to creating the map frame using different methods

4. Refactor the Map Frame Widget
   - Ex: change the implementation of the Map Frame class to allow creating it using different gestures
Extraction Steps

5. Move the Map Frame classes to the API Layer

6. Refactor eGrid classes interacting with the Map Frame
   - eGrid now uses the Map Frame from the API
   - Acceptance tests of eGrid have to pass

7. Use the Map Frame in eWell
   - Usage examples found in eGrid can simplify using the Map Frame in eWell
   - Acceptance tests of eWell have to pass
eGrid

Surface Toolkit

- ScatterView
- ScatterViewItem

BackgroundLayer

GISMap

GhostFrame

Bookmark

BookmarkPin

BckgndLayerMock

Still to be extracted in the next iterations

GISTT_API

MapFrame

- Id
- Orientation
- Center
- Width
- Height

+ InitializeMap()
+ LoadMap()
+ AddLayer()
+ DeleteLayer()

0..* MapFrameMock

0..* MapFrameTest

Tests for the User Stories supported by the extracted MapFrame (supporting both eGrid and eWell)

eWell

AppMgr
Approach Benefits

- Aligns better with **iterative** and **incremental** nature of XP
  - Handles components on demand, one at a time

- Supports reuse **reactively**
  - Minimizes investment upfront
  - Component extraction happens only when needed

- Makes use of existing assets to inform extraction decisions
  - Ex: usage examples and acceptance tests
Challenges

- Extracting User Interface components
  - Creating acceptance tests for these components is tricky – especially if designed using scripting languages
  - Partial solution: using mock classes

- The safety net can also use unit tests
  - Need to be automated to reduce the risks
Next Steps...

- Repeat the process – use the API in other applications
  - Assess the approach further
  - Generalize the API

- Usability Studies of the API
  - Measure the effectiveness and value of the API
  - Assess how useful the extraction approach
Summary

- Extracting **reusable components** and **usage examples** from existing applications can be done in an XP environment using a **systematic iterative test-driven approach for refactoring**.

- We also present a **case study** in which we apply this technique and try to assess its value and usefulness.
Questions