ThoughtWorks STUDIOS

mingle twist go

Agile QA Practices

A course from
ThoughtWorks Studios

Why are we here?

Hopes

Concerns

ThoughtWorks STUDIOS

Course "project" plan

Day 1	Day 2	Day 3
Introduction Review and overview Project inception • Write stories • Assessing quality and progress • High-level testing strategy with team • Release plan	Initiation/Iteration 0 • "Done" for stories, iterations • Get familiar with technology • Plan for iteration 1 Iteration 1 • How and why we automate • Acceptance-test-driven development	Iteration 1 (continued) Iteration 2 • Learn automation practices and patterns • Learn and do exploratory testing Wrapup • Parking lot review • Retrospective

ThoughtWorks STUDIOS

What you will learn

- How agile QA and testing differs from other approaches
- How to develop an agile test strategy
- How to write valuable user stories and acceptance criteria
- Testing practices that work well in agile projects
- How to automate tests

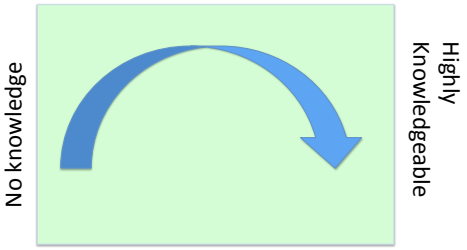
ThoughtWorks STUDIOS

To start…

- Find a partner, so we have some mix of experience – we call this pairing
- Take an index card – write “What We Know” on top
- You have 5 minutes for this

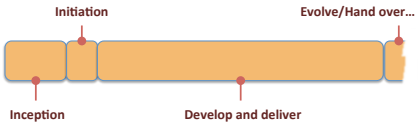
ThoughtWorks STUDIOS

How much agile knowledge?



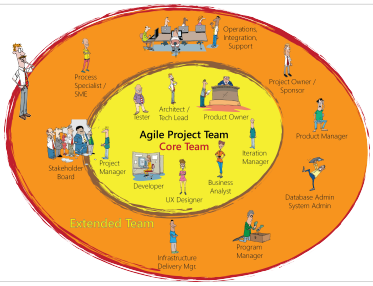
ThoughtWorks STUDIOS

Agile project lifecycle



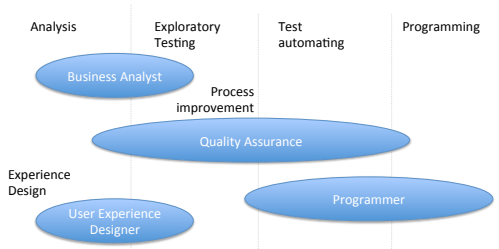
ThoughtWorks STUDIOS

Team Composition



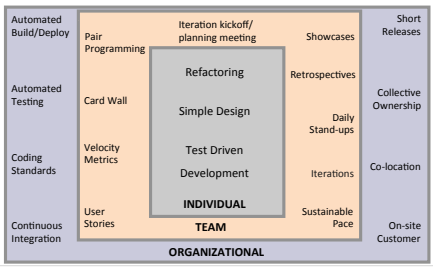
ThoughtWorks STUDIOS

Team activities by role



ThoughtWorks STUDIOS

Agile practices



ThoughtWorks STUDIOS

Practices that benefit the QA

- Unit testing
- Continuous integration
- Pairing and whole-team approach
- Iterative development
- Team co-location and shared space
- Daily standups
- Automated functional testing

ThoughtWorks STUDIOS

The agile *QA* manifesto

How would you pair these like the agile manifesto?

- Testing for risk
- Scripted manual testing
- Detecting defects
- Gatekeeping
- Preventing defects
- Automated and exploratory testing
- Informing
- Testing for coverage

ThoughtWorks STUDIOS

The agile *QA* manifesto

Through this work we have come to value:

Preventing defects	Detecting defects
Automated and exploratory testing	Scripted manual testing
Testing for risk	Testing for coverage
Informing	Gatekeeping

That is, while there is value in the items on the right, we value the items on the left more.

ThoughtWorks STUDIOS

Simplicity, Feedback, Courage, Respect and Communication

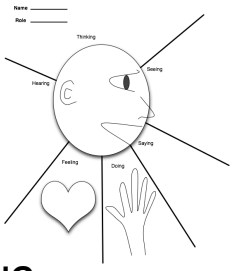
HOW DO THE AGILE VALUES RELATE TO QA?

Crispin and Gregory's tester's bill of rights

You have the right to:

- Ask questions of customers and programmers and receive timely answers
- Bring up issues related to quality and process at any time
- Ask for and receive help from anyone on the project team, including programmers, managers, and customers
- The tools you need to do your job in a timely manner

ThoughtWorks STUDIOS



What's work like now?

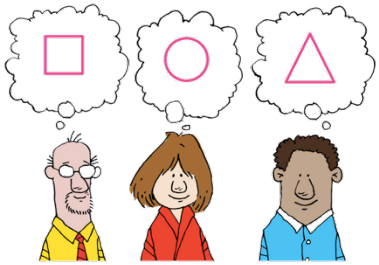
EMPATHY MAPPING

Questions?

Inception

Phase 1 of an agile project

Communication is not always Clear



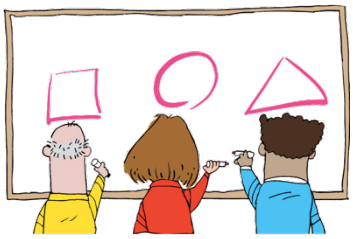
"I'm glad we're all agreed then."

A Shared Understanding

- Why are we doing this project?
- Decision making around scope
- Prioritization of features
- The right solution

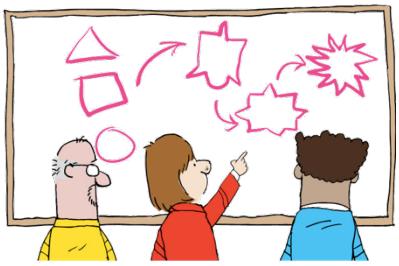
ThoughtWorks STUDIOS

Mental Models



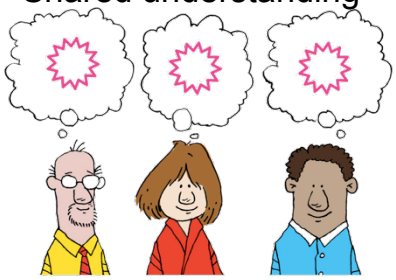
"Ah..."

Convergence



"Ah!"

Shared understanding



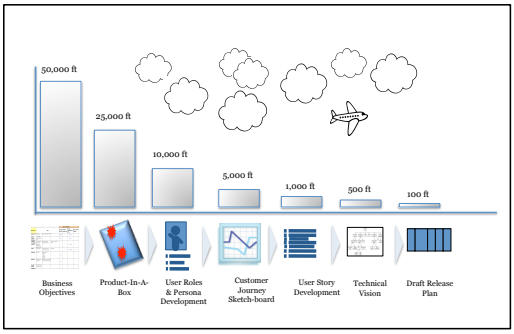
"I'm glad we're all agreed then."

Goals of inception

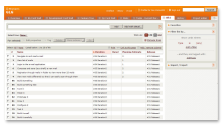
Goal	What QA does
Understand the problem and business context	Help translate business needs into stories
Develop a shared understanding of the scope	Help determine how team will assess quality and progress
Define a candidate architecture	Develop high-level testing strategy with team
Develop a credible plan	Help team create release plan
Build relationships	Be involved with team early and often

ThoughtWorks STUDIOS

Roadmap



Deliverables – The Plan



Initial Story List

Release Plan

Inception task 1

CREATE MASTER STORY LIST

User Stories

Three C's

- **Card**
- **Conversation**
- **Confirmation**

ThoughtWorks STUDIOS

Review: Story structure

- As a <role>
- I want to <goal>
- So that <rationale>

As a Librarian, I want to be able to search for books by publication year so that I can produce a chronological index

ThoughtWorks STUDIOS

Rethinking the role of Agile QA

Traditional Tester Role	Agile QA Role
Is part of separate test team	Is part of the entire team
Testing happens at end of development	Testing happens parallel to development
Works alone	Pairs with BAs, programmers and others
Acts as gatekeeper	Highlights risk
Has no or little contact with business	Has direct contact with business
Tests written and automated after development	Tests are written and automated before and during development

ThoughtWorks STUDIOS

Motivation behind agile testing

- Tests provide the safety net that lets agile projects proceed at a rapid pace.
- Commitment to testing is reflected by the creation of large automated suites and vigilance for any changes that lead to test failures.

ThoughtWorks STUDIOS

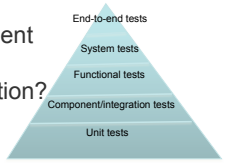
Testing strategy

- Initially defined during project inception
- Light-weight and non-prescriptive
- Outline basic test process
- Types of testing and responsibilities
- List environment and resource requirements
- List dependencies
- Highlight risks and issues
- Strategy can and should change throughout a project

ThoughtWorks STUDIOS

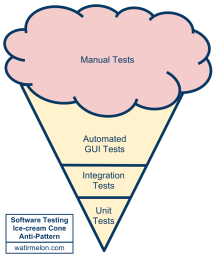
Building a testing pyramid

- How much?
- What should we automate vs. leave manual?
- Which tools?
- When in the development cycle?
- Who does the automation?
- What mix of testing makes sense?



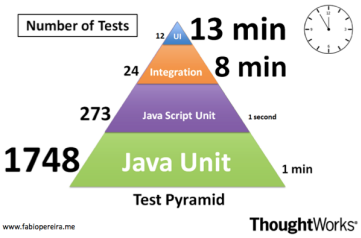
ThoughtWorks STUDIOS

Alister Scott's testing anti-pattern



ThoughtWorks STUDIOS

Testing pyramid: Example



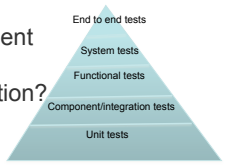
ThoughtWorks STUDIOS

Build your own test pyramid

WHERE WOULD YOU TEST THIS?

Where would you test this?

- How much?
- What should we automate vs. leave manual?
- Which tools?
- When in the development cycle?
- Who does the automation?
- What mix of testing makes sense?



ThoughtWorks STUDIOS

Where would you test this?

Test	Where?	How?
Environment is set up		
Tax is computed properly		
COTS rule engine is configured		
“date of birth” control element works		
Layout of UI is correct		
Capacity (performance)		
Javascript that shows updated number of tickets remaining		
Bulk upload of tickets into new bug tracker (batch process)		
Possible data change in production		
Service is responding		

STUDIOS

Agile testing practices

- Just-in-time test strategy
- Test early: move it forward in cycle
- Automated testing
- Acceptance test driven development
- Manual testing becomes exploratory testing
- “Rightweight” testing

ThoughtWorks STUDIOS

Testing Practices: Test early

- Story card reviews
- Acceptance tests
- Functional tests
- System integration tests
- Performance
- Security

ThoughtWorks STUDIOS

Inception task 2

HIGH-LEVEL TEST STRATEGY

Inception task 3

DETERMINE HOW TEAM WILL ASSESS QUALITY AND PROGRESS

Principles of agile metrics

- Foster transparency, honesty and trust
- Means to the end of improvement
- Use to compare team to itself and not to others
- Use "for a season" to address specific problems
- "Of the team, by the team, for the team"
- Measure team, not individuals
- Big and visible
- Low cost

ThoughtWorks STUDIOS

Questions to ask

Questions to ask	Example areas to track
What is the quality of our product? What is the quality of our process? What is our progress? How good is our testing? What is our truck/lottery number? Where are most of our defects appearing? Are we going too fast? Slow? Where are our bottlenecks	Build performance (time, too many red builds, long-running builds) Stories rejected at showcase Burndown/burnup line Code coverage Pairing habits Velocity consistency Needless interruptions (non-story work) Cumulative flow, Cycle time Work in progress Defects in production Running tested features

ThoughtWorks STUDIOS

Pairamid/Pairing chart

ThoughtWorks STUDIOS

What kind of behavior do we want to encourage?

MAKE YOUR OWN METRIC

Defects

- Stories within iteration don't have defects but are merely done or not done
- Defect fixing is not a parallel stream of work, but part of the main development backlog
- Defects are treated like stories (prioritized and have acceptance criteria).

ThoughtWorks STUDIOS

Initiation (Iteration 0)

Phase 2 of an agile project

Iteration 0 and QA

- Help team determine “done” for stories, iterations
- Get familiar with technology
- Spike an automated test
- Plan for iteration 1

ThoughtWorks STUDIOS

What is “done”?

- Common, objective understanding across team of completeness
- No notion of “code complete”
- Done criteria for stories, iterations and releases
- Only push of a button needs to happen before story/iteration/release is deployed in production
- Developed in conjunction with business

ThoughtWorks STUDIOS

Done criteria examples

- All tests (unit, functional, integration) pass
- Exploratory tested
- Integrated into trunk
- In pre-prod environment
- Demoed to stakeholder
- Meets all cross-functional requirements
- Databases are migrated

ThoughtWorks STUDIOS

Iteration 0 task 1

WHAT IS "DONE"?

Development practices that are useful for QA

GET FAMILIAR WITH TECHNOLOGY

Continuous Integration

"The key is to automate absolutely everything and run the process so often that integration errors are found quickly."

-- Martin Fowler

ThoughtWorks STUDIOS

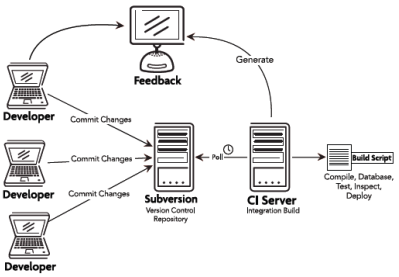
Continuous Integration

Core practices

- Check in regularly
- Create comprehensive automated test suite
- Keep the build and test process short
- Don't check in on a broken build
- Run all commit tests locally after updating, before committing
- Never go home on a broken build (but be prepared when someone *does*)
- Always be prepared to revert to previous revision
- Don't comment out failing tests/assertions
- Visual build monitor

ThoughtWorks STUDIOS

Continuous Integration



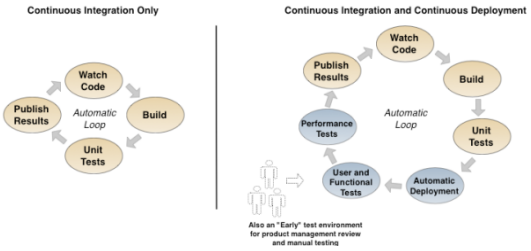
ThoughtWorks STUDIOS

What is a successful build?

- All the latest sources are checked out of the configuration management system
- Every file is compiled from scratch
- The resulting compiled code is put into binaries
- The system is started and suite of tests is run against the system

ThoughtWorks STUDIOS

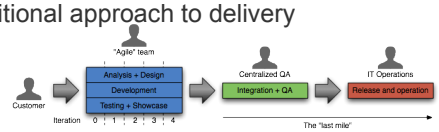
Continuous Integration and Deployment



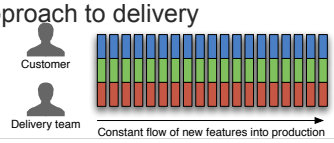
ThoughtWorks STUDIOS

Continuous delivery

Traditional approach to delivery



Ideal approach to delivery



ThoughtWorks STUDIOS

Iteration 0 task 3

SPIKE AN AUTOMATED TEST

Functional test example

Google Test

NCR should be listed on a Google search:

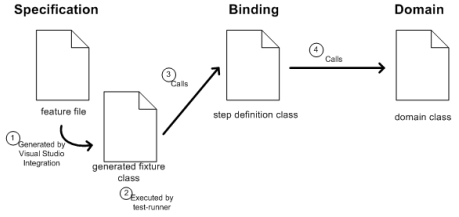
- Given a user is on the Google home page
- When the user searches for NCR
- Then NCR is displayed in the search results

```
// JUnit Assert framework can be used for verification
import net.sf.sahi.client.Browser;

public class NCRShouldBeListedOnAGoogleSearch {

    private Browser browser;

    public NCRShouldBeListedOnAGoogleSearch(Browser browser) {
        this.browser = browser;
    }

    public void givenAUserIsOnTheGoogleHomePage() throws Exception {
        browser.navigateTo("http://www.google.com");

    }

    public void whenTheUserSearchesForNCR() throws Exception {
        browser.textbox("q").setValue("NCR");
        browser.submit("btnG").click();

    }

    public void thenNCRIsDisplayedInTheSearchResults() throws Exception {
        assertTrue(browser.link("NCR").isVisible());

    }
}
```

ThoughtWorks STUDIOS

Functional Testing

Behavior-driven development frameworks



ThoughtWorks STUDIOS

What is Functional Testing?

- Answers the question of "can the user do this" or "does this particular feature work"
- Business-facing, product-facing tests
- Tests that prove you "built the right thing" (as opposed to "built the thing right")

ThoughtWorks STUDIOS

What is Functional Testing?

Types of Functional Tests
- End-to-end testing (i.e., does the application work all the way through)
- Regression tests that reflect a particular defect found in production
- Testing "from a user perspective", which includes acceptance tests
  - Acceptance testing is particularly relevant in Agile
  - Guarantees that the requirement as specified is implemented in the software
  - Some teams use Acceptance test driven development
  - All functional tests are not acceptance tests however

ThoughtWorks STUDIOS

What is unit testing?

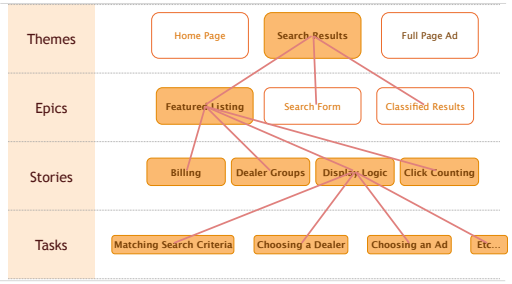
Per Michael Feathers, it's not a unit test if:
- It talks to the database
- It communicates across the network
- It touches the file system
- It can't run at the same time as any of your other unit tests
- You have to do special things to your environment (such as editing config files) to run it

ThoughtWorks STUDIOS

Iteration 0 task 4

PLAN FOR ITERATION 1

Detail is commensurate with plan horizon



ThoughtWorks STUDIOS

Elicit detail just in time

ThoughtWorks STUDIOS

Story attributes

- INVEST
  - Independent
  - Negotiable
  - Valuable
  - Estimatable
  - Small
  - Testable
- 3 C's (Card, Conversation, Confirmation)

ThoughtWorks STUDIOS

Writing acceptance criteria

- Collaborative
- How will you know when we're finished?
- Multiples per story
- All or nothing

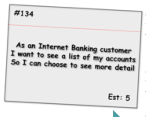
ThoughtWorks STUDIOS

Acceptance criteria

- Given <context>
- When <action>
- Then <expectation>

ThoughtWorks STUDIOS

Acceptance criteria examples



Alternate path

Given the customer has one transaction account and one credit account
When they have completed logging in
Then the screen should show the names and numbers of the two accounts sorted in account number order

Given the customer has just one transaction account
When they have completed logging in
Then the screen should show the name and number of the account

Alternate path

Given the customer has no accounts
When they have completed logging in
Then the screen should show a message stating that no accounts are available

Alternate path

Given the customer has more than 20 accounts
When they have completed logging in
Then the screen should show the first 20 accounts (in account number order) only

Bad path

Given the customer has some accounts
When they have completed logging in
And the system cannot retrieve the account details
Then the screen should show an error message with associated code and details to contact for support

ThoughtWorks STUDIOS

Pre-iteration planning

ADD ACCEPTANCE CRITERIA TO ITERATION 1STORIES

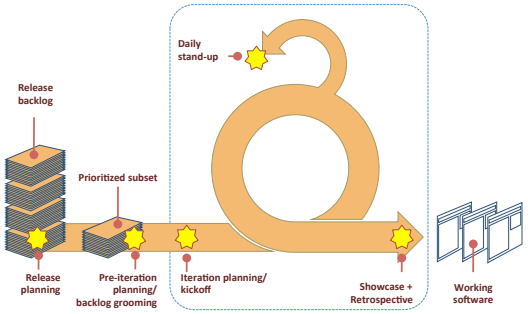
Develop and deliver

Phase 3 of an agile project

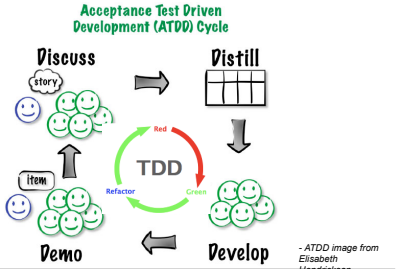
Today's standup

ThoughtWorks STUDIOS

Review: The iteration



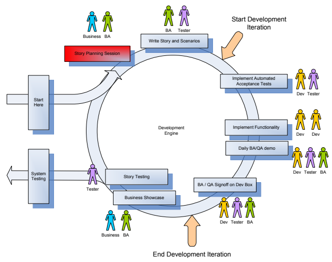
Driving with acceptance tests



- ATDD image from Elisabeth Hendrickson

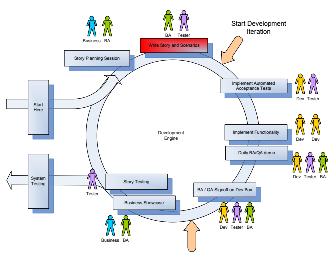
ThoughtWorks STUDIOS

ATDD cycle



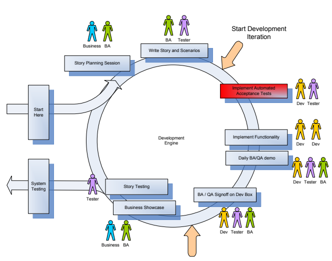
ThoughtWorks
STUDIOS

ATDD cycle



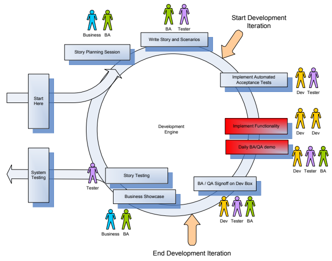
ThoughtWorks
STUDIOS

ATDD cycle



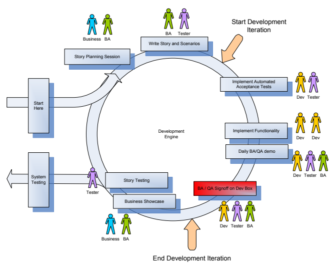
ThoughtWorks
STUDIOS

ATDD cycle



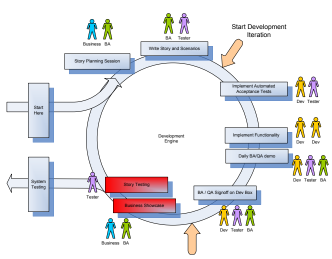
ThoughtWorks®
STUDIOS

ATDD cycle



ThoughtWorks®
STUDIOS

ATDD cycle



ThoughtWorks®
STUDIOS

Functional Testing

- Given <initial context>
- When <action>
- Then <expected result>

ThoughtWorks STUDIOS

Acceptance-testing defects

- Given <initial context>
- When <action>
- Then <expected result>
- *Rather than* <unwanted behavior>

ThoughtWorks STUDIOS

Functional Testing

Automating Functional Tests

- Functional Test Automation Makes Testing an Asset
- Living documentation
- Why Functional Test Automation?
- Unit Testing is Fundamentally Different
- Automated Testing Has Additional "Gotchas"

ThoughtWorks STUDIOS

Why and how we automate

AGILE TEST AUTOMATION

Why automate?

- Provide fast feedback to the team
- Serve as safety net, provide confidence
- Facilitate repeatable testing
- Optimize manual testing
- Verify business case
- Reduce mundane tasks

ThoughtWorks STUDIOS

What's different in agile?

- Automation happens as part of development
- Automate acceptance tests, more than end-to-end tests (in general)
- Automated tests form a regression test suite and are executed in CI to give faster feedback

ThoughtWorks STUDIOS

Who automates?

- Shared responsibility of the team = much more likely to succeed
- Feedback improves the confidence of all team members in the quality of the software
- Shared ownership leads to an understanding of the tests
- Avoids a false confidence in automation

ThoughtWorks STUDIOS

Myths

- Reduce head count
- 100% automation
- ROI from day-1
- Automation script can find more bugs

ThoughtWorks STUDIOS

Develop and deliver

Iteration 2

Today's standup

- Learn automation practices and patterns
- Learn and do exploratory testing

ThoughtWorks STUDIOS

Functional Testing

AUTOMATION PRACTICES AND PATTERNS

Proven practices for test development

- Separate intent from implementation (mechanics)
- Test automation best practices = programming best practices:
  - Abstraction
  - Encapsulation
  - Refactoring
  - Don't repeat yourself

ThoughtWorks STUDIOS

Proven practices for scenarios

- Think (and specify) from the business perspective
- Write tests for reusability
- Refactor steps
- Extract steps as you go
- Rule of thumb for workflow – not more than 10 steps
- Limit verifications and assertions in extracted steps

ThoughtWorks STUDIOS

Proven practices for user-interface tests

UI Changes

– Xpath is difficult to maintain
  Avoid using xpaths

– ID change
  Create constant IDs & reuse

ThoughtWorks STUDIOS

Challenges

Unstable Tests

– Page synchronization
– Waits
– Data dependencies
– Side-effects from tests
– Order/group execution problems

Unclear Tests

– Too many asserts
– Test duplication
– Unnecessary/duplicated setup

ThoughtWorks STUDIOS

Page-object pattern

- Encapsulate all of the actions a user can do or see on a page into a singular object
- A product page would have things like, add to chart, add to gift register, related products, review, etc..
- Typically relies on method chaining – methods in a page object always return another page object
- Doesn't need to reflect entire actual page

ThoughtWorks STUDIOS

Page-object pattern

Positives

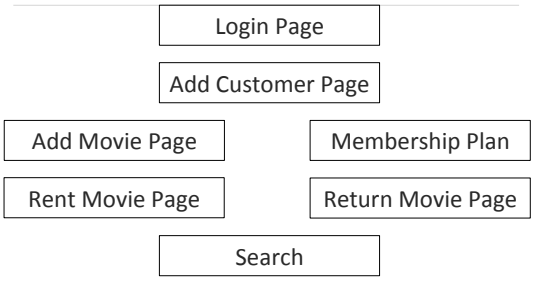
- Makes code more readable
- Makes a navigation map for your tests

Negatives

- Breaks down in a few places
- Assumes that your app page structure is designed well

ThoughtWorks STUDIOS

Page-object pattern



ThoughtWorks STUDIOS

Page-object pattern

```
public class LoginPage {
   public void UserLogin() throws Exception {
     # Login implementation
   }}
public class AddCustomerPage {
   public void AddCustomer() throws Exception {
     # Add Customer implementation
     # enter customer name
     # enter customer ID
     # submit customer info
   }}
public class RentMoviePage {
   public void RentMovie() throws Exception {
     # Rent Movie implementation
     # enter customer ID
     # enter Movie Name
     # submit Rental
  }}
```

ThoughtWorks STUDIOS

Domain-object pattern

- Group all the actions associated with a specific domain concept like search or filtering in the application together.
- Or, groups all the meaningful domain entities together, with relevant actions as methods
- Makes your test suite look more like a library from a code perspective.
- You can implement method chaining, but it requires a map of some sort, which is hard to maintain.

ThoughtWorks STUDIOS

Domain-object pattern

Positives:

- Helps with apps that have many cross-cutting concerns
- Tests don't represent the structure of the application, which is likely to change
- Easy for new users to pick up

Negatives:

- Not as elegant as the Page Object Pattern
- More work to maintain

ThoughtWorks STUDIOS

Domain-object pattern

ACTIONS	ENTITIES
Login	Movie
Add	Customer
Rent	Rental
Return	Membership Plan
Search	

ThoughtWorks STUDIOS

Domain-object pattern

```
public class Customer {
  public void Add() throws Exception {
    # Add Customer implementation
    # enter customer name
    # enter customer ID
    # submit customer info
  }}
public class Movie {
  public void Add() throws Exception {
    # Add Movie Implementation
  }}
public class Rental {
  public void Create() throws Exception {
    # Rent Movie implementation
    # enter customer ID
    # enter Movie Name
    # submit Rental
  }}
```

ThoughtWorks STUDIOS

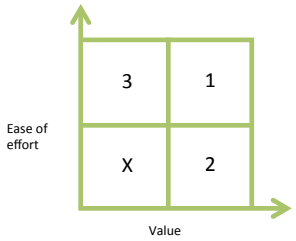
Which is better for our example project?

DOMAIN OBJECT VS. PAGE OBJECT PATTERN

What to automate?

MAKE AUTOMATION CHOICES

Prioritizing tests to automate



ThoughtWorks STUDIOS

What to automate?

- Repeated, scriptable tests
- Tests whose failure feedback is valuable to the project
- Tests for defects
- Tests that aren't extremely complex to orchestrate
- Tests that need to be run often

ThoughtWorks STUDIOS

Why automation fails

- Test code isn't treated like product code
- Over-engineering your test scripts/ framework
- Trying to test everything with functional tests
- Using tools that aren't well suited

ThoughtWorks STUDIOS

Hendrickson's traits of ill-suited tools

- Test-last workflow encouraged by such tools is wrong for an Agile team
- Scripts created by these tools become un-maintainable
- Specialized tools require test-automation specialists and lead to knowledge silos

ThoughtWorks STUDIOS

Characteristics of agile-friendly tools

- Support starting test automation effort immediately
- Separate the test intent from the implementation details
- Support and encourage good programming practices for the code portion
- Support writing test automation code using real languages, with real IDEs
- Foster collaboration

ThoughtWorks STUDIOS

QA-programmer collaboration

- Test code is still code!
- Give the same love and care to test code as well
- Treat test code as a long-lived artifact
- Test code as living documentation

ThoughtWorks STUDIOS

Manual testing complements automated testing

EXPLORATORY AND OTHER TESTING ACTIVITIES

Other testing activities

- Regression testing
- Bug Bashes/Testapalooza
- User-acceptance testing and feedback
- Beta testing

ThoughtWorks STUDIOS

Exploratory testing

- Unscripted, (mostly) manual testing
- Simultaneous learning, test design and test execution
- Supplements scripted automated tests

ThoughtWorks STUDIOS

Cross-functional requirements

- Specify criteria that can be used to judge the operation of a system, rather than specific behaviors
- Define how a system is supposed to *be* (rather than *is*)
- Security testing such as permission/access, penetration, compliance testing
- User-experience testing such as usability, user walkthroughs, A/B, multivariate testing
- Operations testing such as failover recovery, disaster recovery, backup/restore, monitoring, deploy/rollback testing
- A.k.a. “itilities”

ThoughtWorks STUDIOS

For more

Resources and readings:

– http://support.thoughtworks.com/entries/21259508-agile-qa-practices-reading-list

Contacts:

– kmchugh@thoughtworks.com

– rmaurya@thoughtworks.com

ThoughtWorks STUDIOS