

Upfront Test Automation

By

Thanga Srinivasan

Quality Analyst, ThoughtWorks

Agenda

- What is all about?
- Why?
- How?
- Framework Design – Do's & Don'ts
- Demo
- Challenges
- Where can we apply
- Q & A

What is all about?

- Building a smart framework
 - Which enable us to write functional test scripts upfront
- Encourage changes in requirements

Why?

- To deliver the high quality product on/before time
- To avoid redundant manual testing effort from day one
- To certify a build with minimum time and effort
- To help developer change the design or re-factor the code whenever he/she wants

How?

- Doing automation once the part of requirement analysis is done
 - requires a smart automation framework
- QA's needs to know the future changes to the application
 - requires more QA's collaboration with BA's and Dev's
- Need a flexible framework which supports frequent changes
 - requires a simple and smart automation tool

Framework Design – Do's

- Have simple Object Map
- Have simple Test Report
- Have simple Test Data Management
- Have simple Test Suite Management
- Have simple Object Identifying mechanism

Framework Design – Don'ts

- Don't over engineer
- Don't generalize too much
- Don't repeat yourself
- Don't complicate script
- Use acceptable amount of OOP

Demo

Challenges

- Maintenance
 - Refactoring
 - Tagging
 - Implicit synchronization
- Productivity
 - Smart Recorder
- Managing effort
- Coordination with BA's and Dev's
- Need through Tool Knowledge
- Need people who can do both manual & automation

Where we can apply?

- We can adopt this approach in any SDLC models
- Suits very well with Iterative and Prototype models

Q & A

Contact me thanga@thoughtworks.com