

#PNSQC2022

TRI-LAYER TESTING ARCHITECTURE



PÉTER FÖLDHÁZI Jr. (HU)

Quality Architect @ EPAM Systems USA









#PNSQC2022



History

- 2013: first usage of the Tri-Layer Testing Architecture on a project
- 2014: first public mention of the Tri-Layer Testing Architecture
- 2016: ISTQB's 4-Layered Architecture released
- 2020: first standalone presentation of the Tri-Layer Testing Architecture
- 2021: STARWEST in person event cancelled 😕

#PNSQC2022

- 2022: finally sharing it on the STAREAST conference!
- And now with you all \odot







Spoiler Alert!!!





<epam>

TEST AUTOMATION ARCHITECTURE

Framework Expectations



#PNSQC2022







Framework Expectations

AUTOMATED TESTING

DETAILED LOGGING

GENERATED REPORTS

CI/CD INTEGRATION

EASY TO USE

LOW MAINTENANCE







https://www.peterfoldhazi.com



Using the Right Term

- Test Automation Architecture (TAA)
 - High-level architecture design of the TAS
- Test Automation Solution (TAS)
 - Realization of a TAA
 - Includes Test Automation Framework
 - Integration Test & Project Management
 - Integration to Configuration Management
 - CI/CD pipelines, environments etc.
- Test Automation Framework (TAF)
 - Base of the TAS

#PNSQC2022

• Test Harness, Test Libraries, Test Scripts & Suites



Framework Layers





Framework Layers

- Layers define distinct border of classes that have similar purposes
- Many people tend to overuse layers

#PNSQC2022

• This brings in unnecessary complexity





https://www.peterfoldhazi.com



Integration Into Software Systems

- Typically, simple frameworks have only one SUT (System Under Test)
- E2E testing may involve validation of multiple applications and services
- E2E testing typically includes different types of testing
 - UI, API, Database, Analytics etc.

#PNSQC2022



https://www.peterfoldhazi.com





Multiple Frameworks

- Test Automation Solutions sometimes include multiple frameworks for different purposes
- Typically, different levels are covered with different frameworks
- Some of the base libraries could be shared amongst these frameworks



eng

BI-LAYER TESTING ARCHITECTURE

Unstructured / No Architecture





#PNSQC2022



Unstructured / No Architecture

40TH ANNUAL

WARF



epam>









Bi-Layer Architecture





#PNSQC2022



Bi-Layer Architecture





#PNSQC2022

<epam>





#PNSQC2022





Tri-Layer Architecture

40TH ANNUAL ≩ SOFTWARE

≚ CONFERENCE

OCTOBER 10-12, 2022

ALII



Test Scripts Layer

- Test Scripts
 - Runnable scripts
 - Implementations of all the "ready-to
 - automate" processed test cases
 - Each test step shall directly be called from facades provided by the flow models

https://www.peterfoldhazi.com

- Test Suites
 - Set of automated test scripts

#PNSQC2022

• Typically: @Smoke & @Regression







Business Logic Layer

- UI Test Automation
 - Page Object Models
 - Flow Models
- API Test Automation
 - Utility classes (GET, POST, PUT, DELETE)
 - End-points
 - Param, URI, Request builders
- Common Business Logic
 - App specific BaseTest

#PNSQC2022

- App specific validations, loggers
- Test user and other data configuration

Business Logic		
App Base Test	Flow Models	Page Models



https://www.peterfoldhazi.com

Core Libraries Layer

- BaseTest, BasePage, BaseUtil
 - No application dependency
 - Basic functionality with base tool
 - Any team and project may use it
- Project Common Libraries
 - Project dependent
 - Domain specific functionality
 - Only project related apps may use it
- Additional Libraries
 - Solution dependent

#PNSQC2022

• Solution specific functionality



Examples: sensor mocks, parsers, loggers





Summary of Layers





<epam>



N

10/

- One native Android app development team under Project #1
- One test automation engineer

#PNSQC2022

• Designs the architecture using the Tri-Layer concept

Project #1		
App #1 Test Scripts		
App #1 Business Logic		
Core Libraries		





- Project #1 kicks-off another Android team
- Same test automation engineer working on both apps
- Reuses all the application independent libraries and saves couple of weeks' work



https://www.peterfoldhazi.com



#PNSQC2022



• Project #2 kicks-off an Android team

#PNSQC2022

- A second test automation engineer joins that team
- First engineer provides the core library for the second engineer who becomes productive already in Sprint 1







- Project common libraries are also stored in the Core Layer
- These libraries could span over multiple projects, and it's not mandatory for everyone to include those libraries





https://www.peterfoldhazi.com



Repositories

- Test & Business Layers of an app should be in the same repository
 - They may even be in the same repository as the SUT (Software Under Test)
- Core Libraries should be in the same repository
- New external utilities can be integrated as dependencies in the Core Layer
- Common libraries only used for a set of teams/projects should be handled as separate modules



Modernization

- You may swap libraries to new or different ones in the Core Layer
 - Does it affect any teams negatively?
 - If yes, then only make the change inside your boundaries, therefore in your Business Logic Layer
- You may even swap build tools

40TH ANNUAI

• In the Business Layer you may and should override Core functionality (BaseTest, BasePage, Assertion etc.)





Case Study - Architecture



Case Study – Coding Example

40TH ANNUAL ≩ SOFTWARE

≤ CONFERENCE

#PNSQC2022





Summary and Conclusion

- The Tri-Layer Architecture provides a simplified way of designing Test Automation Frameworks through improved abstraction
- There's an enhanced reusability of framework components
 - Application independent libraries can be reused in any project with similar testing goals
 - Project specific libraries can be reused in multiple teams within the same project
 - Costs of initial test automation efforts are decreased
- Collaboration is encouraged and communication is improved between colleagues or even amongst engineers globally through open-source development



InflectraCon 2022 in Washington D.C., USA





#PNSQC2022



https://www.peterfoldhazi.com



40TH ANNUA



#PNSQC2022











References

- Special thanks to:
 - Gergely Ágnecz, Adam Auerbach, Philip Soffer, Andrew Pollner, Patrick Quilter, HTB (Hungarian Testing Board), EPAM North American QA Consultants
- HUSTEF 2014 Péter Földházi Jr. Effective Mobile Automation:
 - <u>https://www.youtube.com/watch?v=I2Y7VGI-iYE</u>
- ISTQB CTAL Test Automation Engineer 2016 syllabus:
 - <u>https://www.istqb.org/downloads/send/48-advanced-level-test-automation-engineer-documents/201-advanced-test-automation-engineer-syllabus-ga-</u>
 <u>2016.html</u>

https://www.peterfoldhazi.com

- My personal website
 - <u>https://www.peterfoldhazi.com</u>

#PNSQC2022

- GitHub example project
 - <u>https://github.com/foldhpet/flowmodelpatternjava</u>





Valley of Fire

Palace of Fine Arts

Yellowstone



Redwoods

#PNSQC2022







