

Testing Stories: From IBM Host to ReactJS

Sriram Ilango

s.ilango@zurich.com

QA Meetup, Trivago 22.06.2023



About me

- I like to test things
- I get paid to do that since 2014
- Test Automation Engineer @ Zurich Insurance Group, Germany
- I talk a lot about *what to test* instead of *how to test*



Software Testers in 2023 - Starter Pack

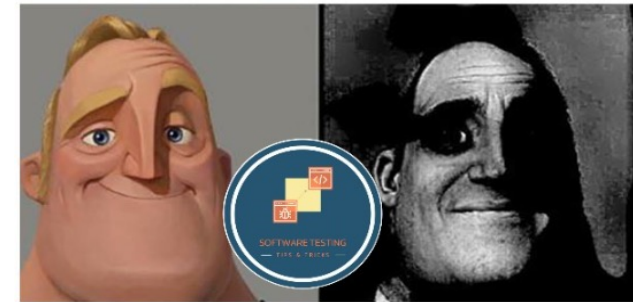
.. always on the lookout for new tools



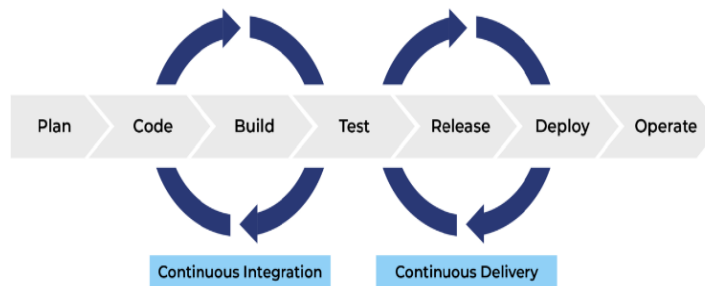
..Love writing Tests than Maintaining them

Writing
automation
tests

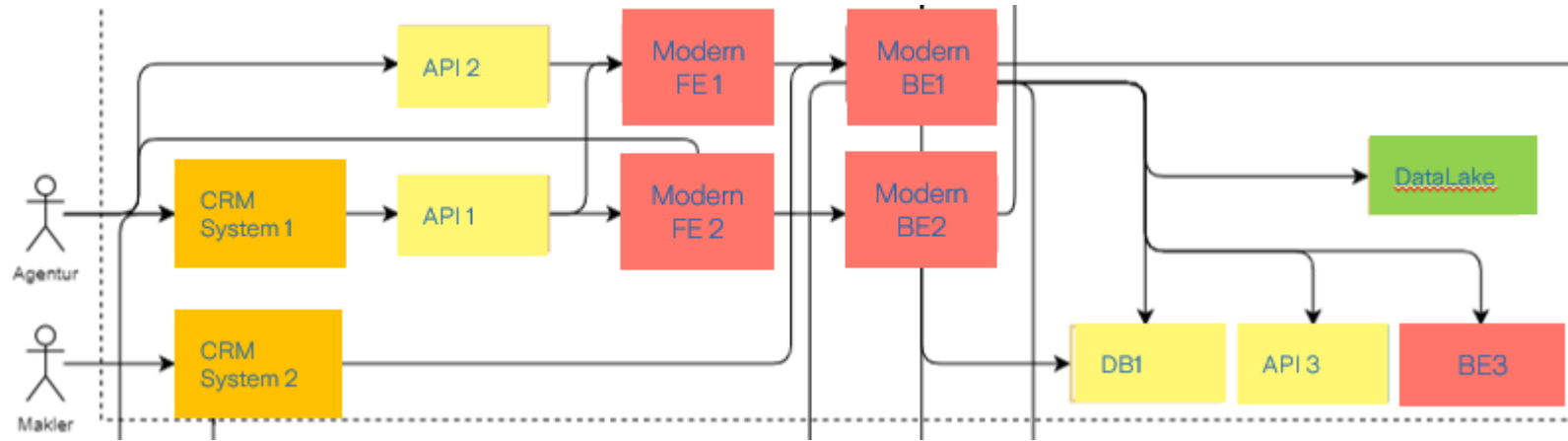
Maintaining
automation
tests



.. Are you even a Tester if you don't run your Tests in a Pipeline?!

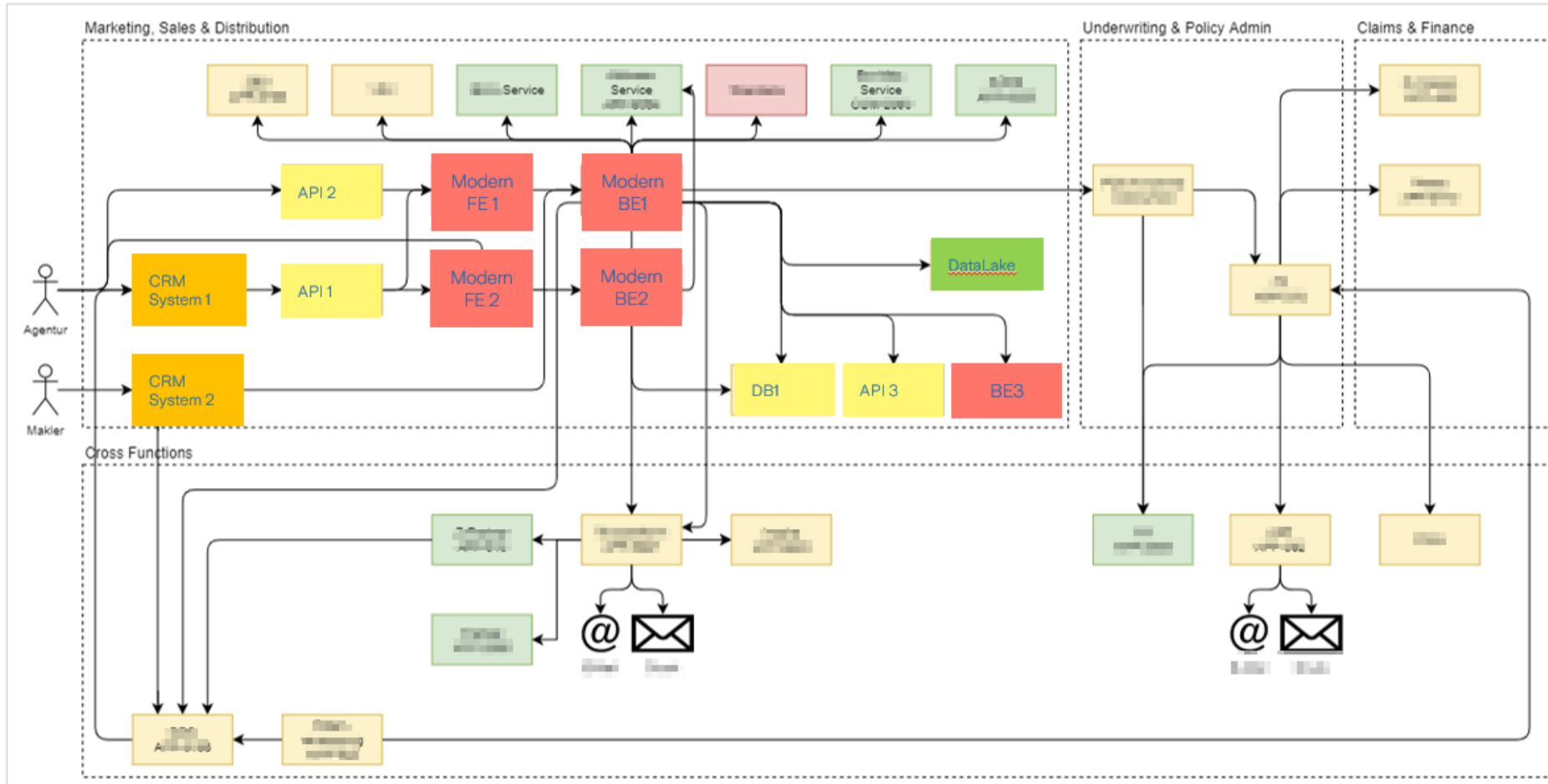


The things we test @ Zurich



We are API-First, agile, deploy Software with zero downtimes and test at all stages in our pipeline.

The things we also test @ Zurich



Some of these systems were built in the 90s and 2000s using technologies like IBM Host, Smalltalk, Vaadin, PrismFlow and IceFrames.

Turns out there is a name for this – “Two-Speed IT”



Fluid

Agile & iterative app
frontend development



Rock Solid

Stable & compliant access
to enterprise data

“borrowed” from: <https://www.progress.com/blogs/gartner-is-right-enterprises-need-bimodal-it-a-k-a-2-speed-it>

Testing Tales from the Mainframe Universe

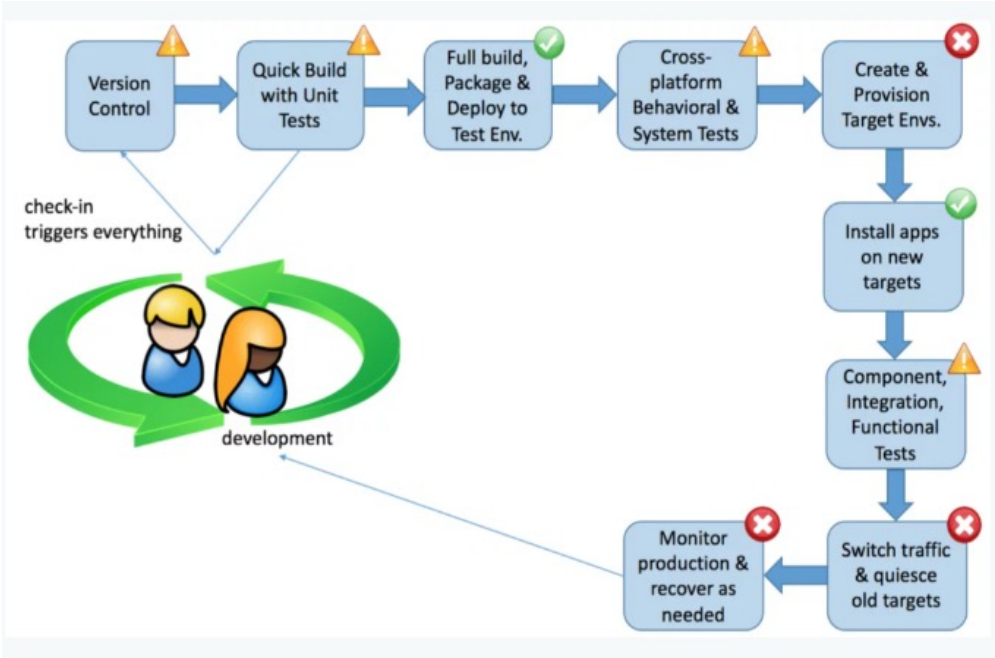
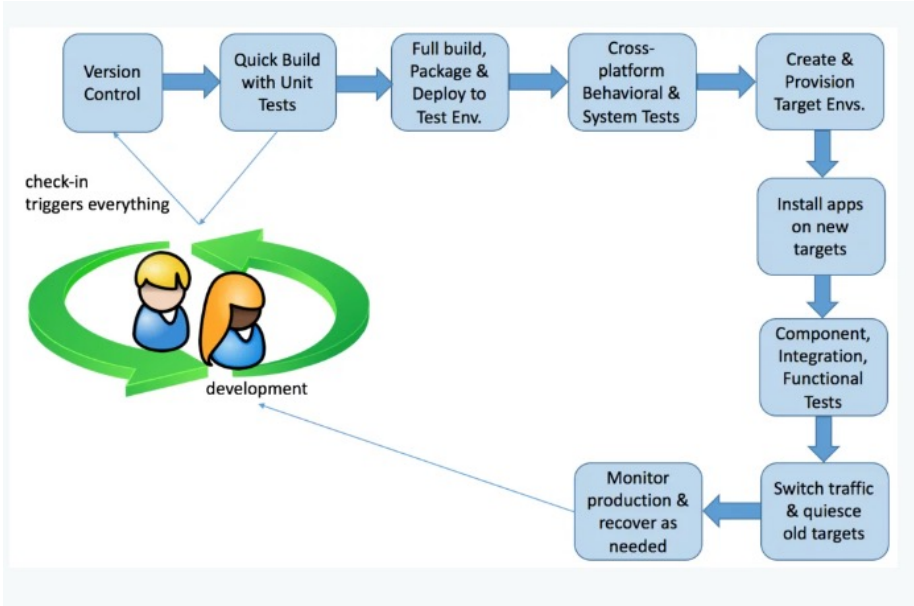
Some stats about modern Mainframes

- In 2022, it was estimated that mainframes
 - power around 70-80% of the world's financial transactions.
 - process up to 90% of the credit card transactions in use
 - process around 30 Billion Transactions a day
- Mainframes are ridiculously reliable and resilient.
- Due to the range of clustering, failover, and disaster recovery solutions, some Mainframes have never experienced downtimes.

Then where is the Problem?

- The problem is that Mainframes work, and we don't change things if they work.
- Testability –
 - there are no *easy ways* to automate System Tests or Regression Tests
 - there are no *easy ways* to rollback or deploy a different code
 - there are no *easy ways* to setup test environments

What does a normal deployment process look like in 2023?



Almost every single stage we take for granted while deploying a modern application is difficult with Mainframes, especially Testing.

Image Source: <https://www.leadingagile.com/2016/11/the-state-of-mainframe-continuous-delivery/>

How did we 'modernize' our Mainframe Tests

Consider Mainframe Terminals/Old Frontends as just another platform.

Write Tests, run them in pipelines just as other distributed systems

The relatively new Frontends of Mainframes are still old Windows based Applications. Use libraries like Winium/Karate to leverage the modern features to test Windows Frontends.

Modernize our tooling for Automated Testing by leveraging Process Automation Tools for Test.

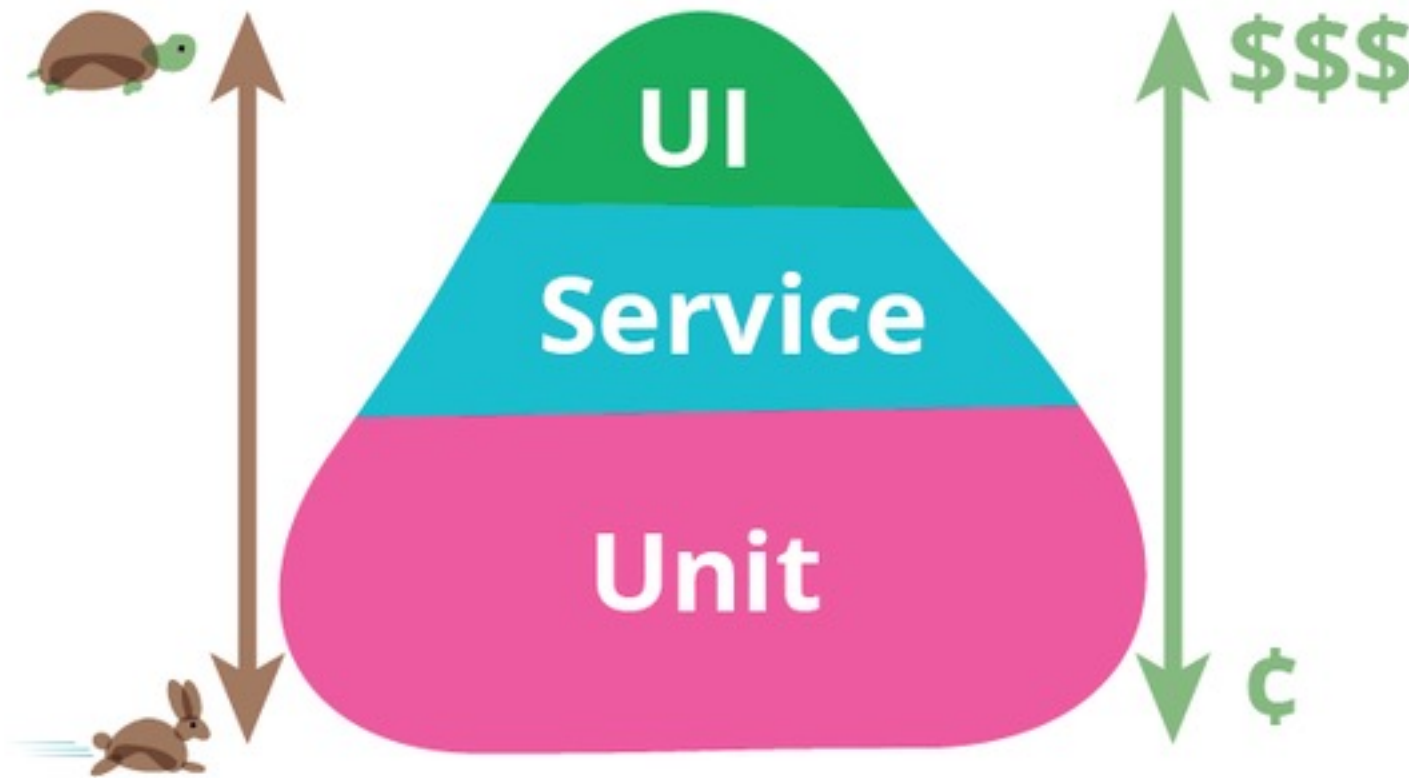
Example: BluePrism

Talk with our Mainframe Testers and find out where they are stuck.

The power of a fresh set of eyes lies in its ability to challenge assumptions and detect blind spots

The Test Pyramid and a story of Deceptive Assurance

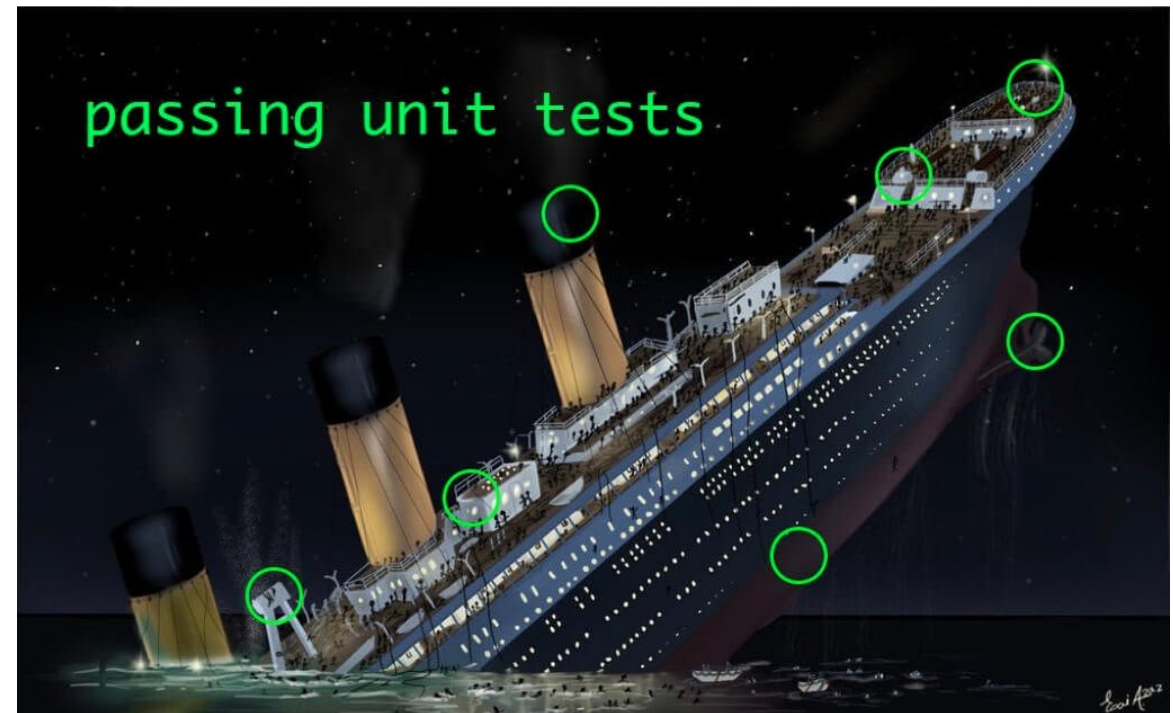
The Test Pyramid



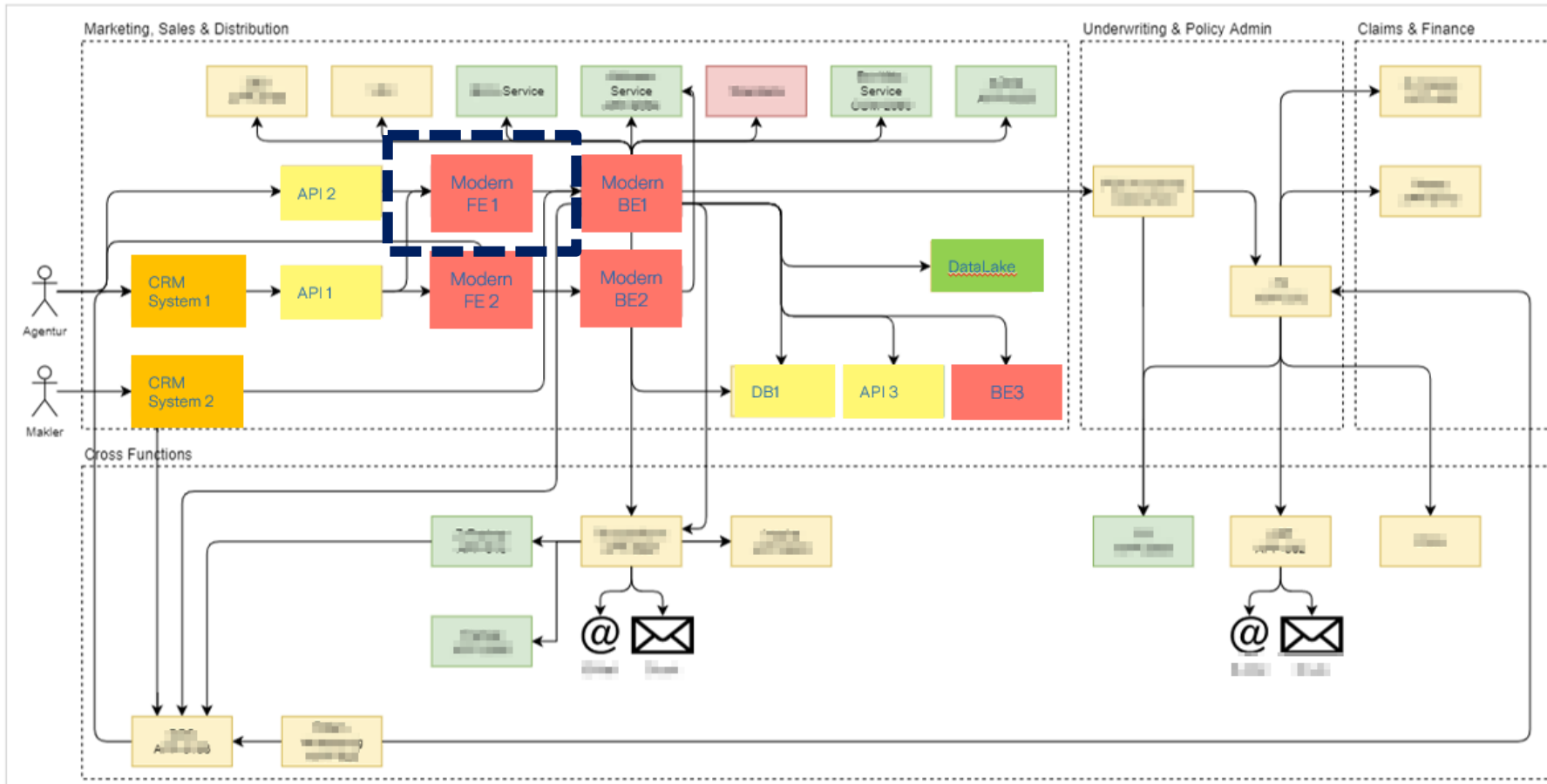
This Pyramid, first coined by Mike Cohn and later popularized by Martin Fowler in his 2012 article is almost the equivalent of the bible for Software Testers.

The article: <https://martinfowler.com/bliki/TestPyramid.html>

All my 9000 Unit Tests passed!



The things we also test @ Zurich



If you are writing Unit Tests for your JS based Modern Frontend 1, would your IDE even allow you to code, compile and deliver a broken component?

And even if you do, would probably mock or stub your services which are controlled/developed by a completely different Team.

Assumptions that need to be challenged about UI Tests

- UI Tests are always brittle and slower
- UI Tests run for a long time, often days or weeks together.
- UI Tests take a long time to develop

These assumptions were usually true in 2012 partly because of the capabilities of the tools we had in 2012.



Selenium RC



You don't have to take my word for it

- Martin Fowler himself claims that the Pyramid is based on this following assumption:

“

The pyramid is based on the assumption that broad-stack tests are expensive, slow, and brittle compared to more focused tests, such as unit tests. While this is usually true, there are exceptions. If my high level tests are fast, reliable, and cheap to modify - then lower-level tests aren't needed

”

- Selenium 4.0 was released in 2022.
- Chrome has had a stable version of their “headless” browser for almost 6 years now.
- There are many wonderful open-source frameworks like Karate that let you automate *any* type of test including pre-built features like Reporting, Assertions and CI Integration.
- You also have JS-based-developer-friendly UI automation tools like Cypress and Playwright that are very capable.
- Even the enterprise tools now days are not bad.
- No tester I know runs their tests locally anymore.

How did we 'mainframize' our modern tests

In addition to the Unit Tests, we run our Service and UI tests in the deployment pipelines.

We also run these Tests across Environments – DEV/SIT/UAT parallelized to reduce run time

Increased focus on Reliability: We try to make our systems more reliable by including load and performance tests often within the sprints

We also run the Customer Facing UI Tests across different devices and platforms all through the CI/CD

Challenged our own assumptions as to what a Service/UI must or must not do.

Collaborate and learn more about the applications we test

- Two-Speed IT is not a long-term solution. But that doesn't mean Testers from different worlds cannot learn from each other.
- No matter what system it is that you are testing, having Automated Tests is crucial.
- As Testers, we need to challenge the trends, assumptions and conventions in our industry.
- There is no “one correct way” to test a Software - Every system is different – all that matters is you catching them Bugs, as early as possible.

The text 'Gotta catch 'em all!' is written in a blue, bubbly, hand-drawn font with a white outline. It is set against a rectangular background with a grey and white checkerboard pattern.

Thank you!

Sriram Ilango

LinkedIn: <https://de.linkedin.com/in/sriram-ilango-28411614>

Email: s.ilango@zurich.com

