

The logo for Exigen, featuring the word "exigen" in a lowercase, sans-serif font. To the right of the text is a square icon with a white diagonal line from the top-left to the bottom-right. A registered trademark symbol (®) is located to the right of the icon.

exigen[®]

Unlocking Value of The Business™

Care for quality, not for bugs!

Do you know the difference?

Ainars Galvans





*The best tester isn't the one who **finds the most bugs** or embarrasses the most programmers.*

*The best tester is the one who gets **the most bugs fixed.***

Cem Kaner

Postponed

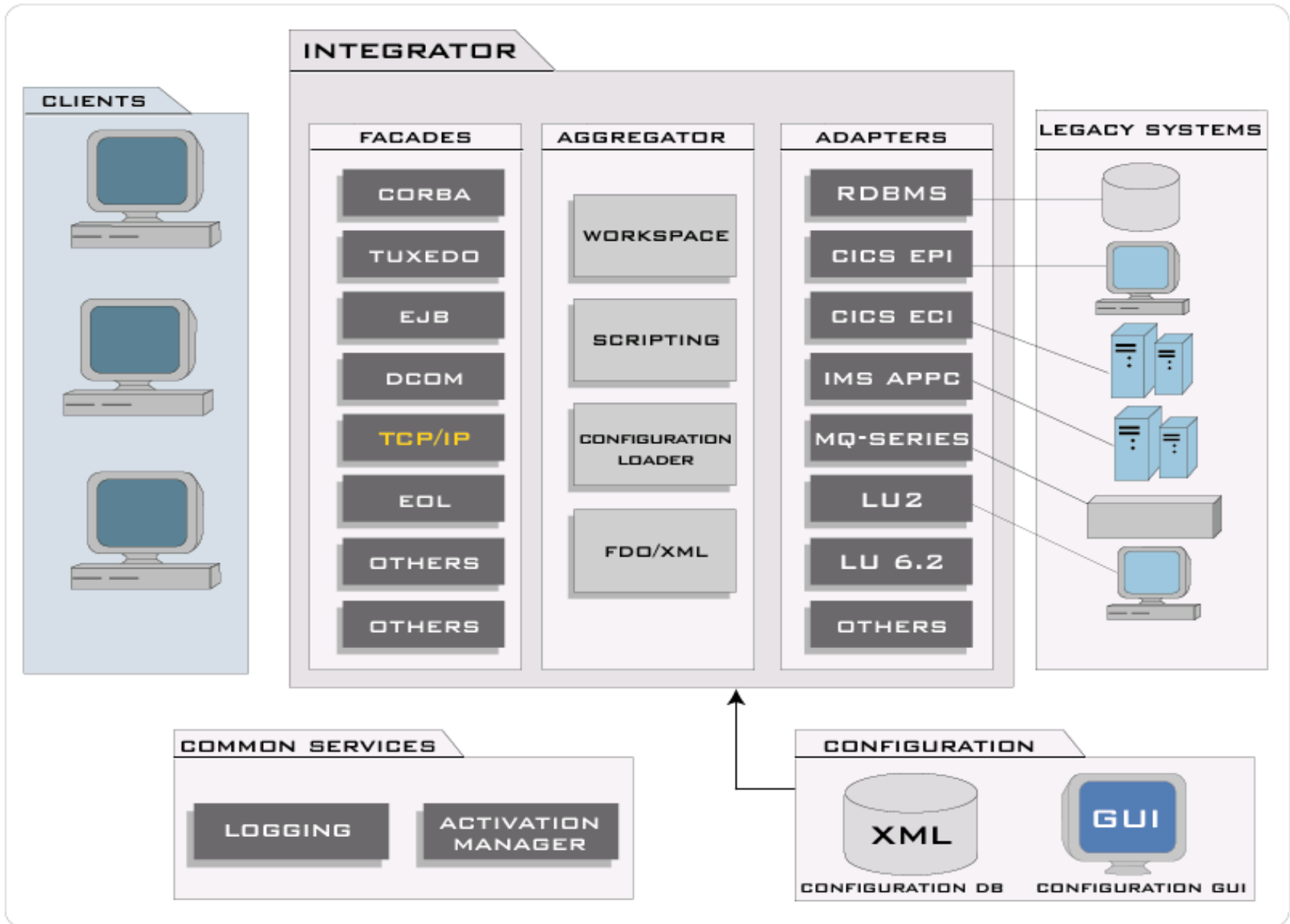
Duplicate

Not a bug

Won't fix



- Project size: 3 testers, more developers
- Software: server with UI for it's configuration
- Project:
 - Adding more features on demand
 - Improving usability of configuration UI
- Context
 - My second team to lead
 - In parallel – doing master in university
- Career impact: earned reputation of
 - Smart and hard working geek
 - Able testing leader
 - With academic background

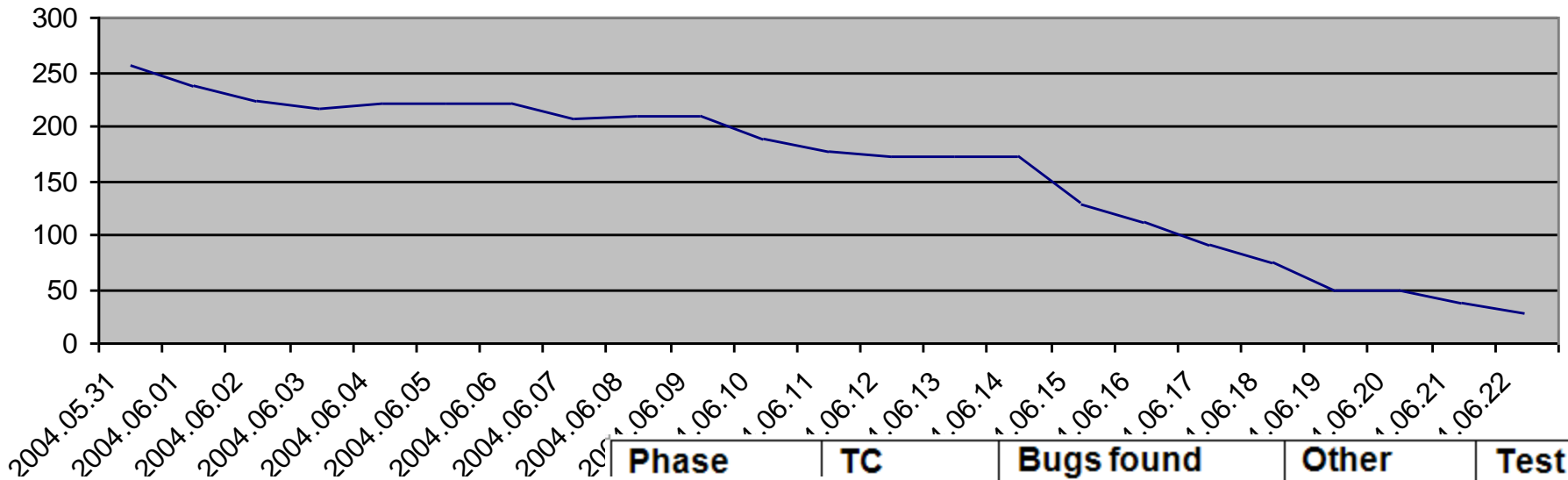




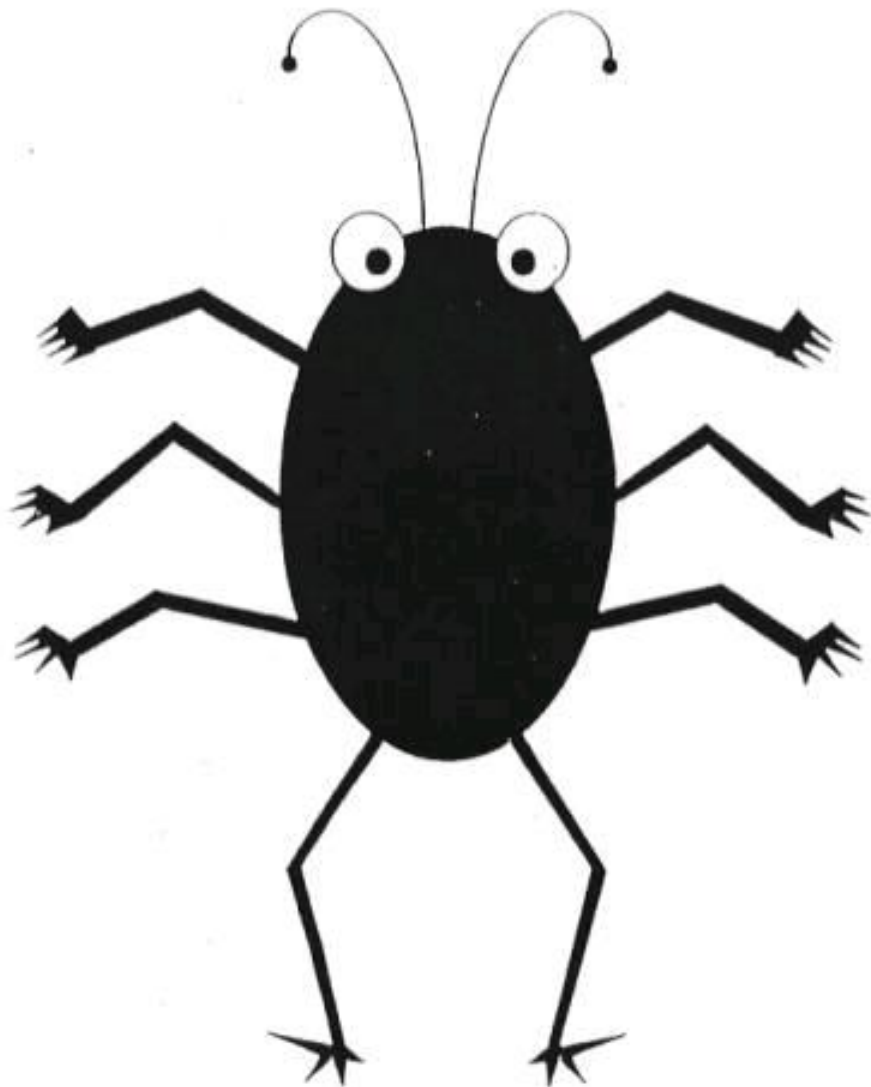
- How I earned my reputation – the Scenario
 1. Worked **hard**, volunteering for extra tasks
 2. Became **SME** of the software
 3. Unofficially did 3rd level **support**
 4. Learned how to choose **what not to test**
 5. **Invested** saved **time in test automation**
 6. **Surprised** everyone when automation done
 7. **Shared** my success at TAPOST conference
- The drawback:
 - Became **irreplaceable** – meaning no career growth/change



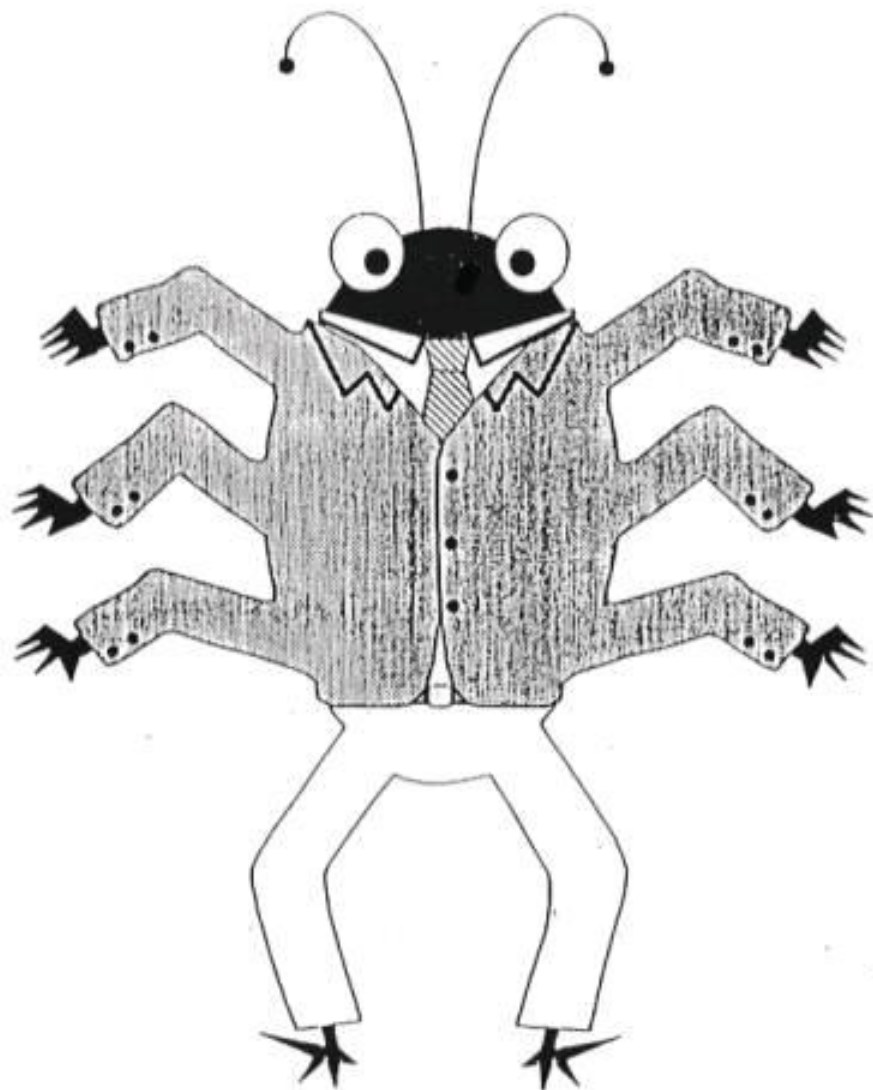
Open defects



Phase	Run-Logs	Bugs	TC	Bugs found during test execution	Other bugs found	Test cycle
Configuration	340	57		398	250	08/25/2
Final Regression	173	8		62	90	11/24/2
Solution Regression	13	2		104	39	02/19/2
Other Activities (e.g. bug retesting)	N.A.	43		80	129	02/19/2
				73	43	05/31/2



BUG



FEATURE

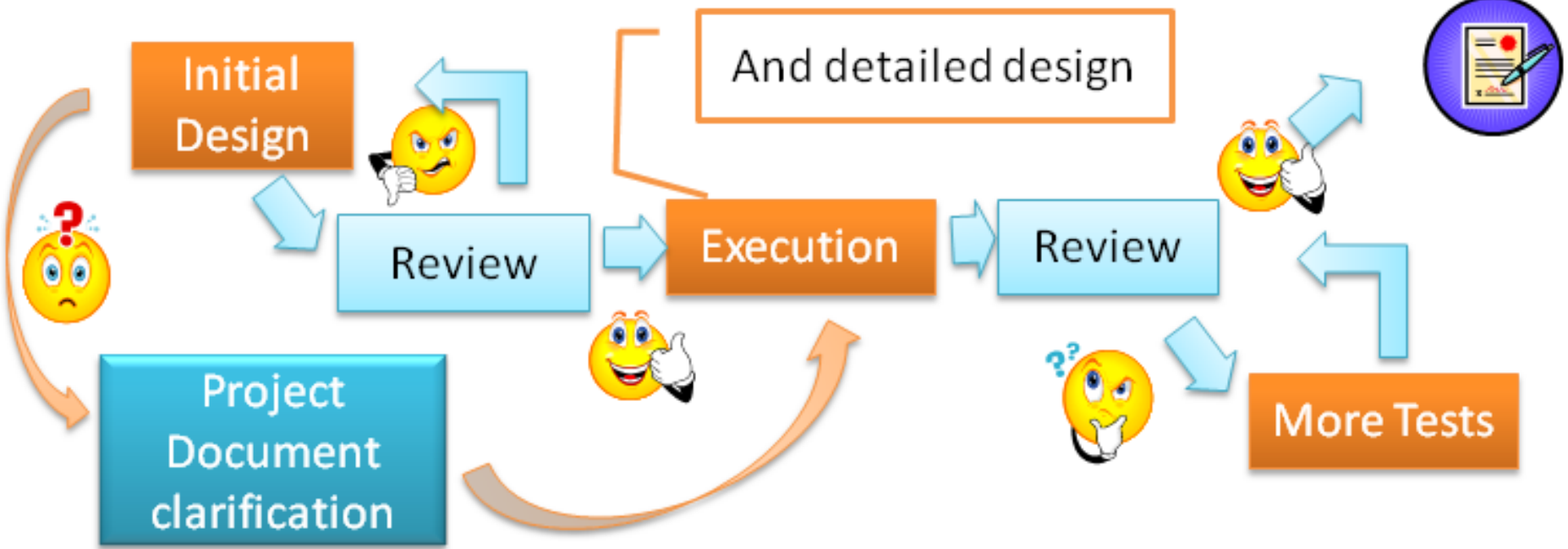


- Root Cause Analysis experience
 - We analyzed about 100 customer bugs that were fixed during a year
 - We spent about 30 person-days to do this
- Results
 - We found nothing we didn't know before
 - We were proud of our engineering as less than 10% had root cause there (development/testing)
 - We learned how to do RCA and why it's so expensive if done properly
- Summary: we were probably even too effective testing what's possible to test



- Context
 - Pilot project (R&D investment)
 - Supposed for internal usage within company

Test Design Life-Cycle





- Ship new feature
 - You want to get feedback as soon as possible
 - You may “improve” it later
 - Because of a risk: you may have to rework it
- Improving feature later may be hard
 - Customer may reject any changes even good ones (office 2007)
 - Management may push for more new features
 - It may break your automated tests



(value: requirements working well enough)
Business



Tester
(value: tests done)

Developer
(value: features)

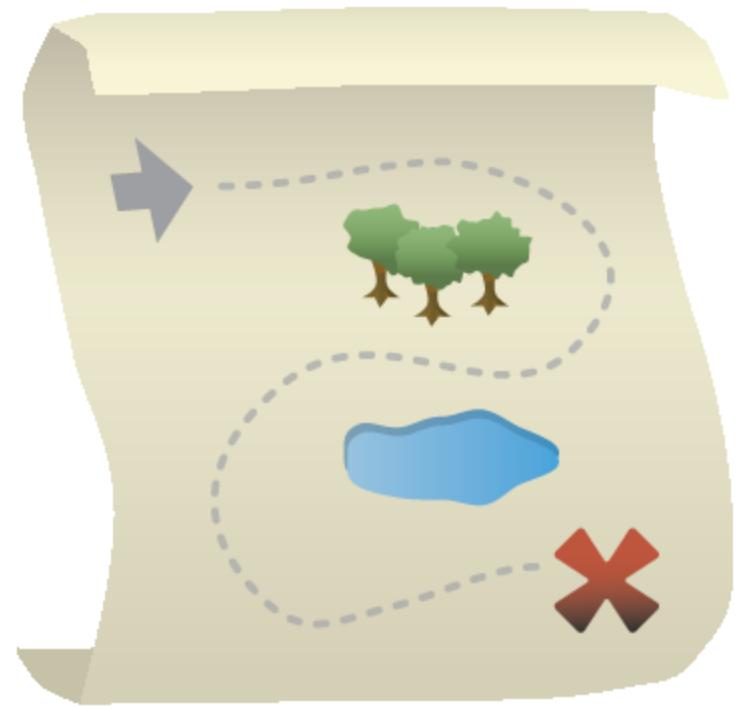




- Requirements based
- Exploratory Test
- Management

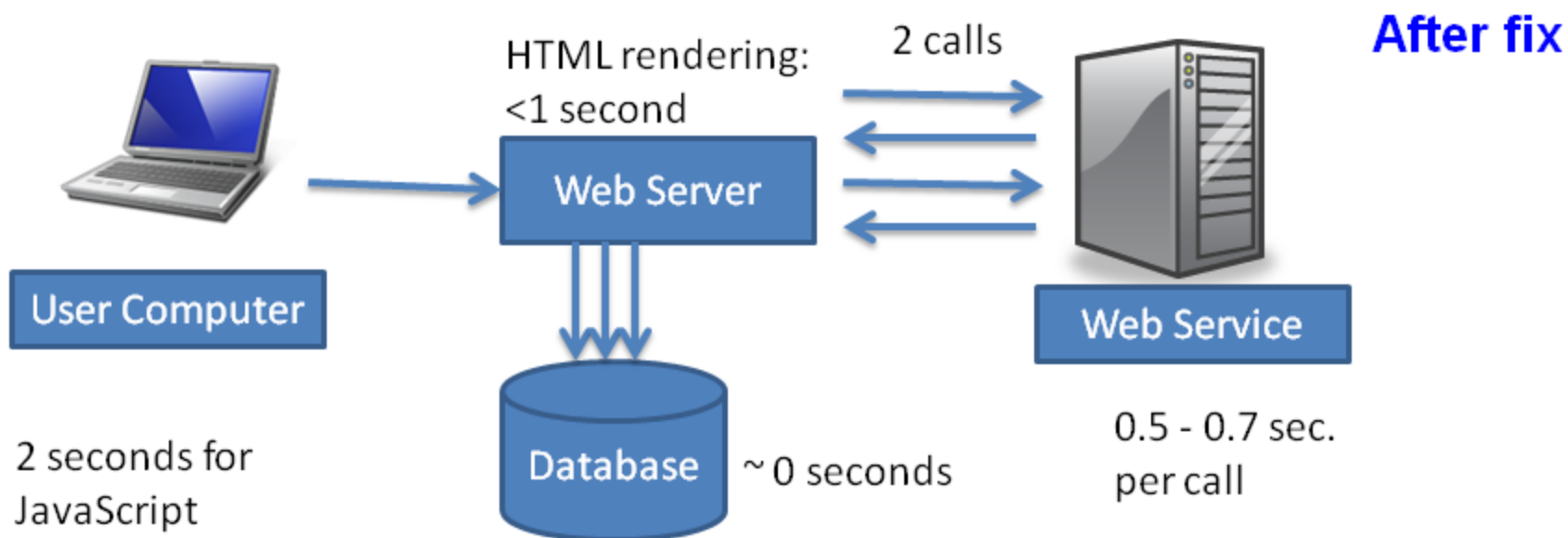
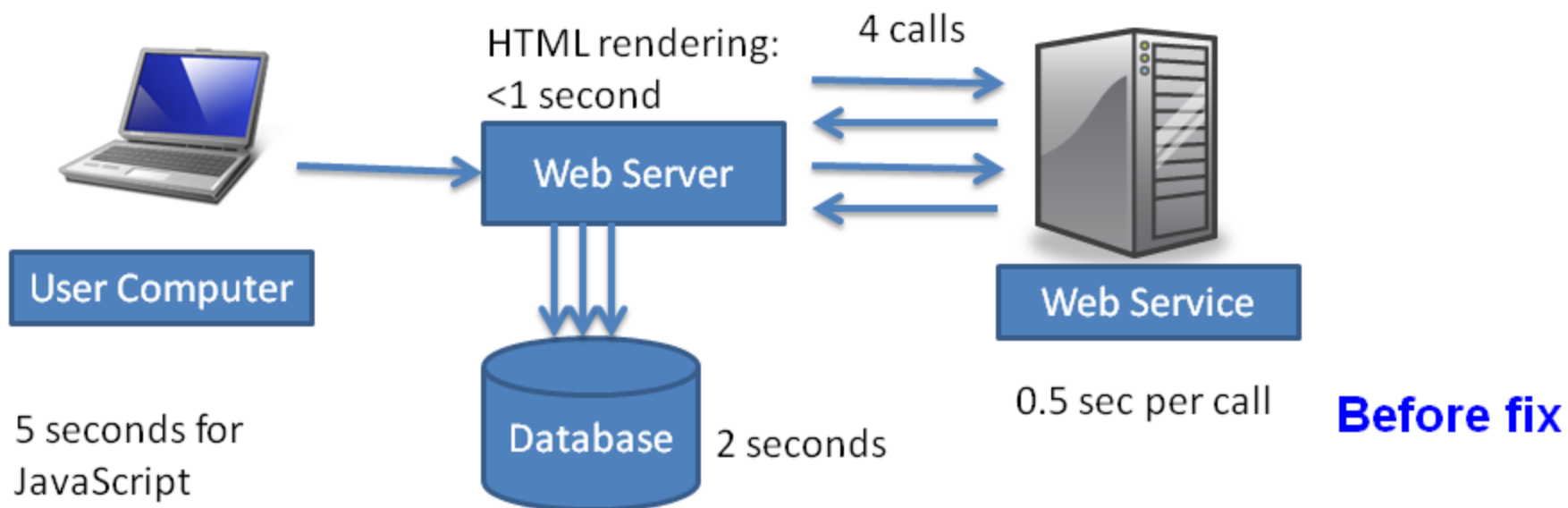
- Test documentation is for tester's use only

- Test Case = "Requirement item"
 - No test steps
 - Only status: pass/fail/not yet
 - Status marked despite of workarounds





- Requirement: response time less than 5 seconds
- Actual during first tests: 9-12 seconds
- Optimization effort (including my time):
 - Client side optimization: 2 months
 - Server side architecture change: 4 months
 - Database caching and indexing: 2 weeks
- Communication with business
 - Increased minimal “hardware requirements”
 - Increased 5 to 7 seconds for this – the most complicated screen of the whole system





- Fixing a bug means:
 - Effort (to decide, to fix and to test)
 - Risk (to break something else)
- Depends on consequences of not fixing
- What are your heuristic Oracles:
 - Heuristic: “It could have been done better”
 - How much to improve usability and performance?
 - Is your code as bad as original requirements?



Skype name: ainarsgalvans

Phone: +371 29432698

Facebook: Ainars Galvans

Email: Ainars.Galvans@gmail.com

Blog: find me at SoftwareTestingClub.com

