Testing Enterprise Software Rewrites

Fri, 1/12/2024 4:00 pm — Sandusky, Ohio, United States

Umang Nahata

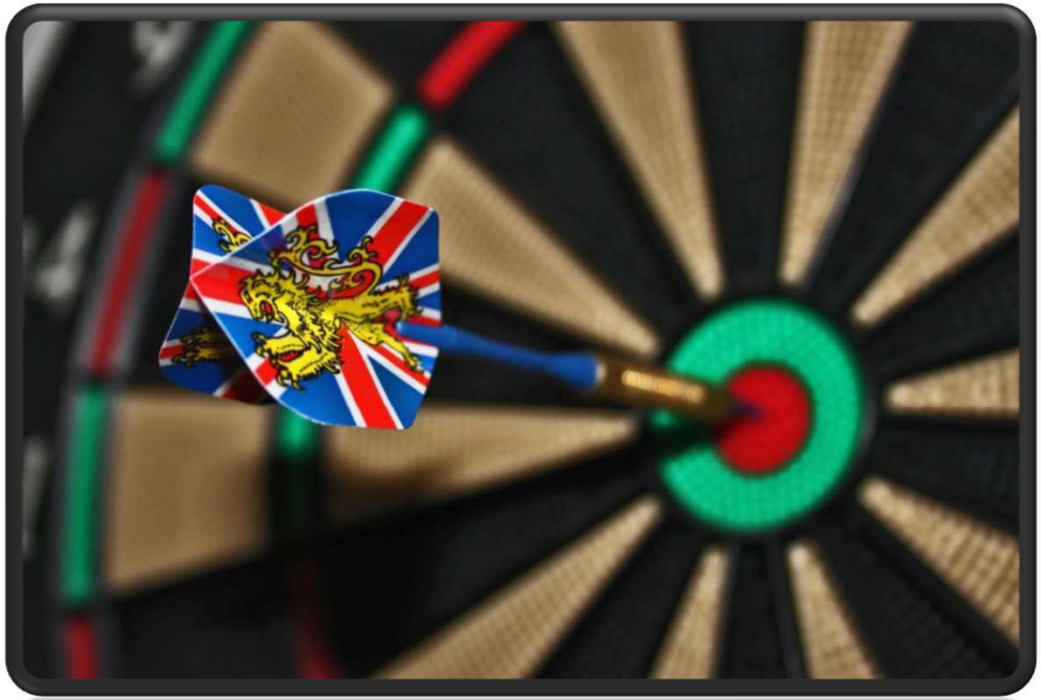
2024 CODEMASH PRIME

Hello!

- Systems Test Engineer (STE) - Lead
  - Progressive Insurance – Ohio
  - Hybrid Developer / QA
  - Create tools and processes to solve testing challenges
  - 15+ years in IT
  - Avid learner

Session Goals

- Enterprise Software Rewrite
- Assessments, Strategies and Experiences
- Practices and Lessons Learned
- Questions and Answers

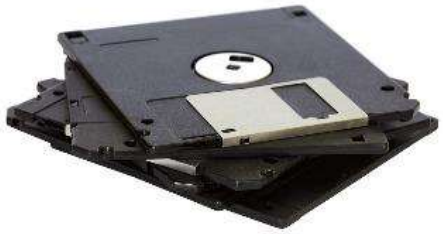


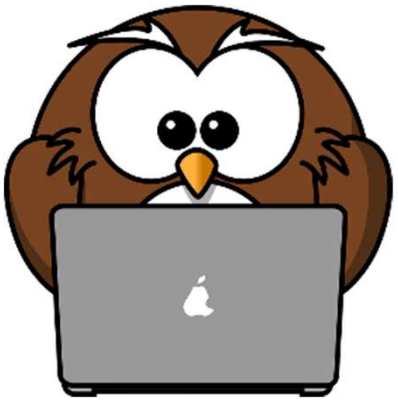
Enterprise Software Rewrite

- Software Modernization Effort
  - Keep up with industry standards, technologies and infrastructure
  - Triggered by different drivers of change
- Rewrite Core Objective
  - Ensure functional compatibility of the new system before retiring legacy system
- Rewrite != Redesign

Enterprise Software Rewrite

- This effort is
  - Risky
  - Costly
  - Challenging
  - Rewarding

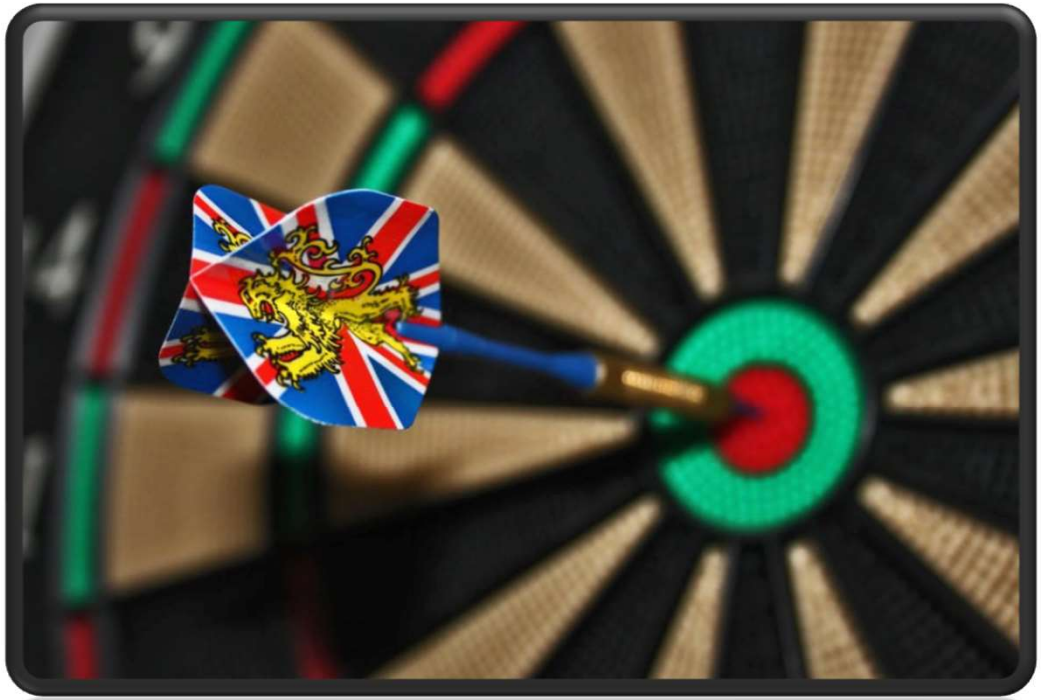




Sometimes, for some enterprises, software rewrite is the only option…

Session Goals

- Enterprise Software Rewrite
- Assessments, Strategies and Experiences
- Practices and Lessons Learned
- Questions and Answers



Know yourself and your customer

“If you know the enemy and know yourself, you need not fear the result of a hundred battles.”

- Sun Tzu, The Art of War

Know yourself and your customer

- 5 Why's Technique ~ Understand dependency and root cause, Explore cause-and-effect

Know yourself and your customer

Why are we doing this? – To implement ‘xyz’

Why do we need to implement ‘xyz’? – To modify feature ‘abc’

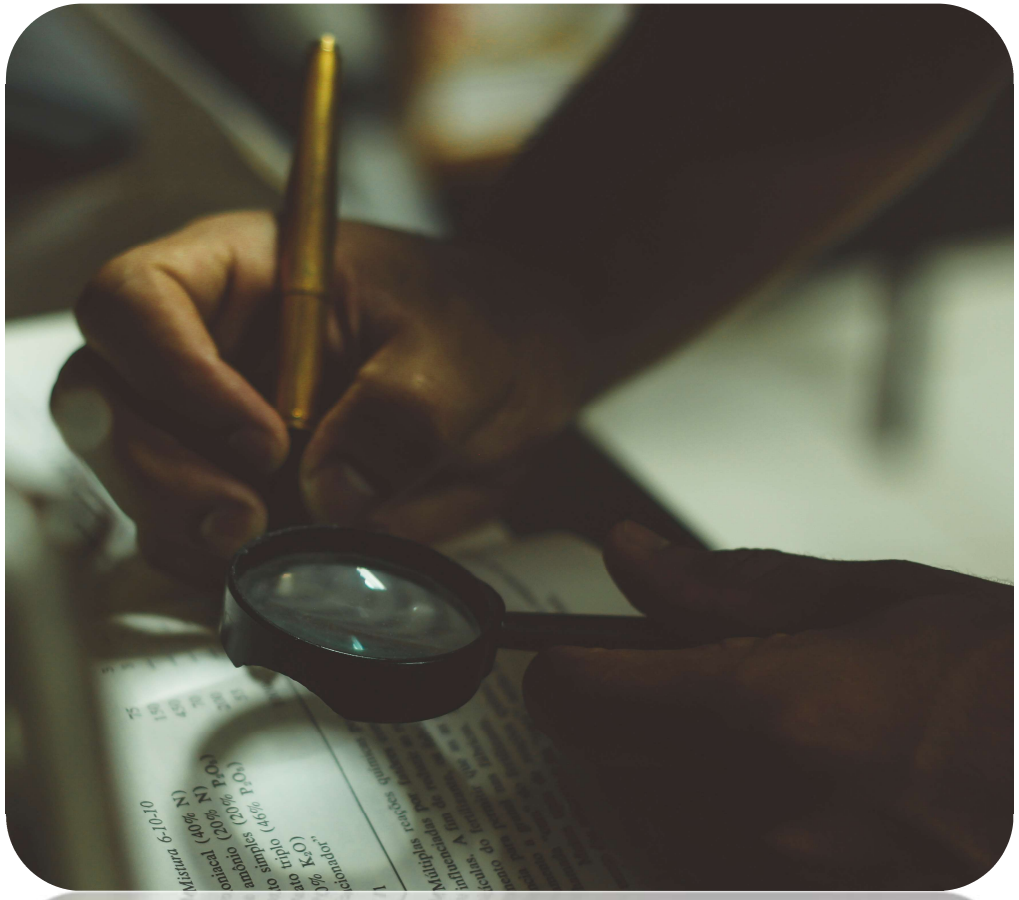
Why do we need to modify ‘abc’? – To avoid maintenance costs

Why? – Vendor retiring support

Know yourself and your customer

5W1H Technique ~ Information gathering

– Who, What, Where, When, Why, How



Know yourself and your customer

Additionally, try to understand

- Change Driving force
  - Strategic, Financial, Infrastructure, Others
- Team Competency
  - Right people, Right skills, Right Attitude, Right time
- Legacy system (as best as you can)

Customer goals != Project goals != Individual goals

Risk Analysis and Management

- Worst Case Scenario
  - What can go wrong? – Description
  - Possibility of it going wrong? – Likelihood
  - How bad can it go wrong? – Severity

Risk Analysis and Management

- Helps plan, manage, allocate resources for testing efforts
- Enterprise software rewrites are high-risk
  - Never underestimate complexity (and hours)
  - Careful of scope creep & feature change requests (How would you test?)
  - Risk for Bread-and-Butter application?

<u>Assess, Adjust and Learn</u>

Experience and Strategies

1. Minimize software architectural transformation
2. Utilize legacy system as test oracles
3. Effective Automation can make a difference
4. Audit test data for internal systems

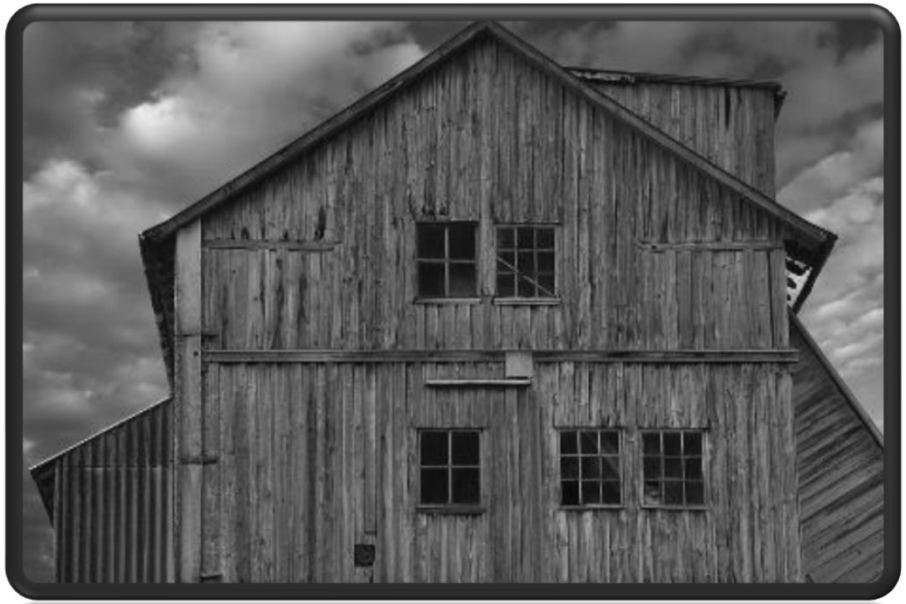
Minimize software architectural transformation

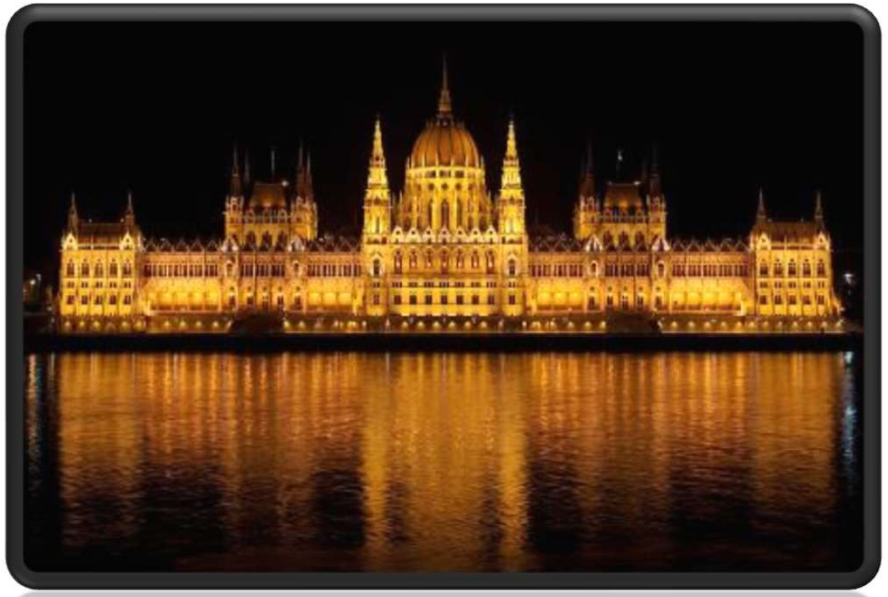
- For effective delivery, scoping and testing
  - Keep it simple and incremental
  - Focus on Minimum Viable Product (MVP)
  - Isolate changes, dependencies (if possible)
  - Resist over design/architect
  - Scope creep could affect testing

(Minimum viable product)

(Final product)

If it (Software Rewrite) seems too good to be true, it probably is…



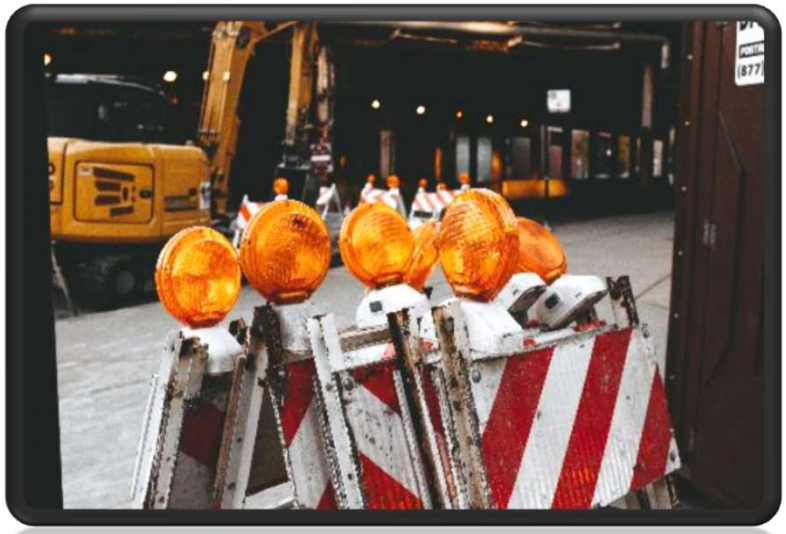


Utilize legacy system as test oracles

- Your testing needs to be strategic
  - Expect lack of Documentation, Requirements, Specifications and Support
  - Baseline your tests on legacy system and compare to new
- Difficult to shoe-horn tests into legacy system
  - Value, Time, Costs, Skills Trade-offs?
  - Methodology ~ Waterfall versus Agile?
- Possible to write tests in new code base
  - Please do (Your team will thank you later)

Effective Automation Can Make a Difference

- Automation
  - Hinder or Support?
  - Right Team, Right Tools, Right Skills, Right Time?
- Determine
  - Purpose & Value
  - Feasibility
  - Reusability

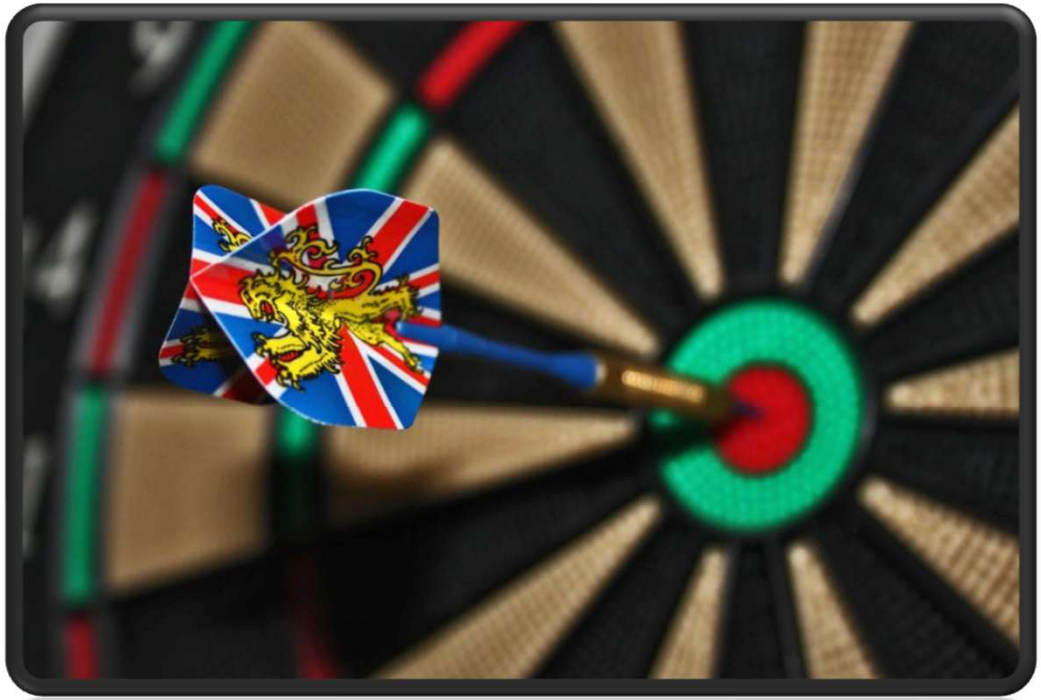


Audit test data for internal systems

- Optional or Adapt
  - Production test data availability?
- Testing the test data
  - Quantity Vs. Quality
  - Repeating same tests?
  - Sufficient test sets? Representation?
- Code Analysis
  - Coverage gaps?
  - Redundant/Unreachable code?

Session Goals

- Enterprise Software Rewrite
- Assessments, Strategies and Experiences
- Practices and Lessons Learned
- Questions and Answers



Teamwork makes the difference

- You may not get it right the first time. It's okay…

- For your team
  - Establish Goals
  - Create environment
  - Encourage mistakes
  - Encourage learning
  - Provide support
  - And more…

Rewrite with specific purpose/target

- Can’t abandon legacy-code
  - Don’t underestimate complexity
  - Try to Reuse/Refactor
  - Focus on building Minimum Viable Product
    - Shows progress to stakeholders
    - Easy to test as well
- Add unit tests when rewriting
  - Think about Automation Pyramid
  - Document as you rewrite

Include people familiar with the legacy system

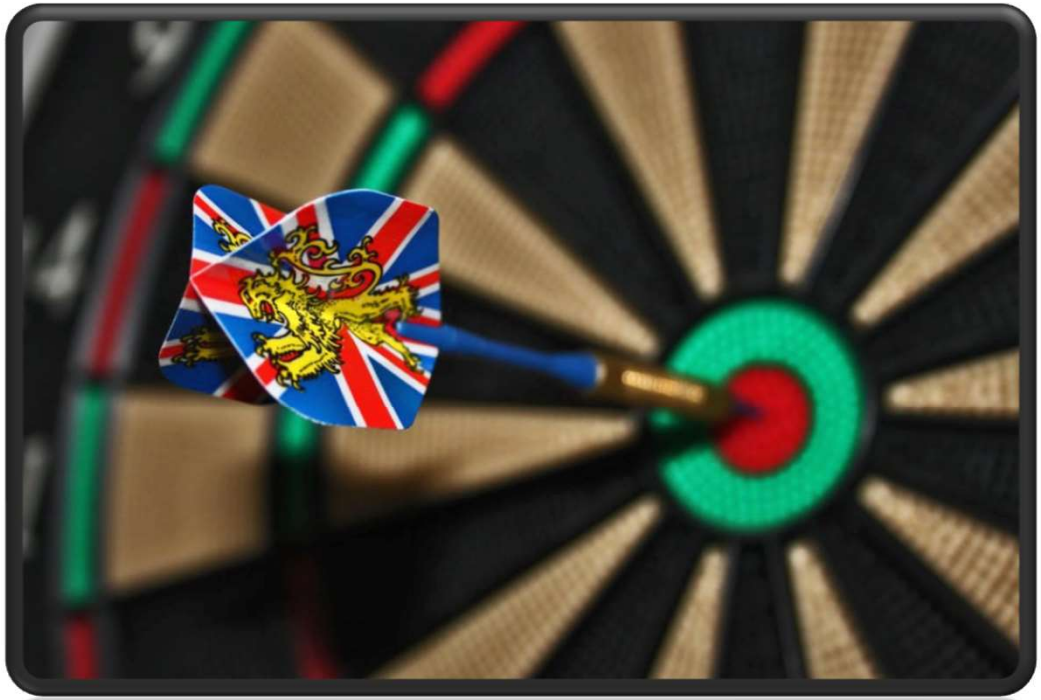
- Need dedicated people and resources who are familiar with old system and include them in rewrite efforts
- These individuals understand how the legacy system interacts, impacts other components and its integration
- Blend a team with old and new technology experts
  - Include them from the start and negotiate response times (if needed)

Remember…this isn't a silver bullet

Testing relies on feedback…
So do I…Thank you!

Session Goals

- Enterprise Software Rewrite
- Assessments, Strategies and Experiences
- Practices and Lessons Learned
- Questions and Answers



Questions and Answers

Appendix – Resources

- Wikipedia – Rewrite, Software Modernization, 5_Whys, 5W1H
- Joel On Software – Things you should never do
- Michael Feathers – Working Effectively with Legacy Code
- Richard Bender – Testing Software Rewrites
- YCombinator – Discussion Link
- James Shore – Surviving a software rewrite
- Stephen Haunts – Is it a good idea to write tests for your legacy code

Appendix – Automation Pyramid