ThoughtWorks®

Custom Software and Knowledge Management

SECURING YOUR ORGANISATION'S KNOWLEDGE DNA, AS A CRITICAL BUSINESS ASSET

ram@thoughtworks.com

www.sriramnarayanan.com

@sriramNRN

Document expectations in executable form along side the software source code to preserve knowledge and development history.

You want this

- No dependency on the individuals who created your software.
- Ability to repair/improve what you paid for

By doing this

- Acceptance Criteria as Executable Tests
- Tests and Code together

Not this!

- When someone leaves, they take the know-how with them!
- Pay exorbitant money to a vendor to fix obvious issues for years together!

Typical (but Avoid!)

- Manual Testing
- Tests in Documents
- QA separate from Dev

TYPICAL SOFTWARE DEVELOPMENT

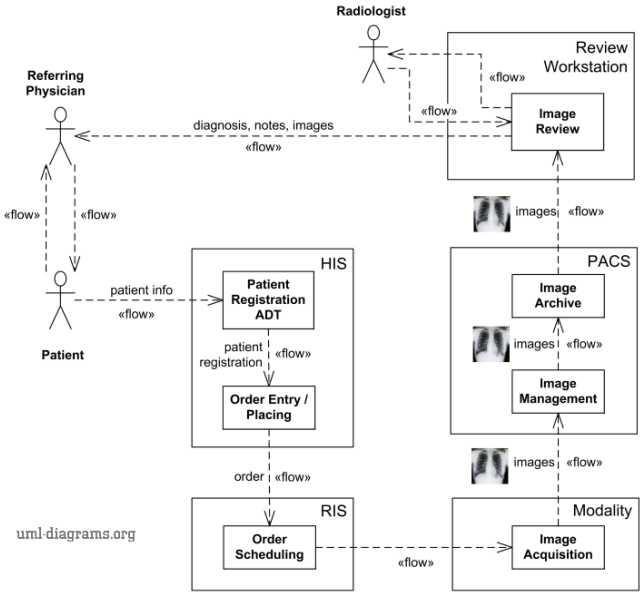
Expected

- Everything documented to specifications
- Everyone understands everything
- “Vendor will take care of everything!”

Reality!

- Changes to leverage opportunities
- Documentation updated later
- The actual developers go away someday…

TYPICAL DOCUMENTATION APPROACHES



UML Diagrams

BLUEPRINTS

FLOWCHARTS

WHITEBOARD DRAWINGS

Image Credits:
http://www.uml-diagrams.org/information-flow-diagrams/scheduled-workflow-example.html?context=ifl-examples
Free Stock Photos

Common issues

- Diagram to code – not always a match
- Diagram to code – prone to mis-interpretation
- Updated after all development is completed
- Often not completed
- Often out dated
- Documentation not by those who wrote the code
  - Incomplete, not factual, not reflective of the reality
  - Considered a “boring task”

Common issues

- “Documentation is out dated, I’ve to reverse engineer”
- “Documentation is incorrect, we’ve to reverse engineer”
- “The technology doesn’t work as per the diagrams – I’ve to do something else”
- “The developer / QA/ BA who knew this workflow left years ago”

IS YOUR ENTERPRISE SOFTWARE LIKE THESE?

A Marvel

No one knows how they were built

Can’t easily be rebuilt or modified

Many unknown aspects

Image Credits: Wikipedia

IS YOUR ENTERPRISE SOFTWARE LIKE THESE?

Rickety Structure

Risky to live in

You wouldn't dare to walk into one!

Watermelon Green

“Green on the outside, red on the inside!”

ThoughtWorks®

WORKING BETTER WITH SPECIFICATIONS

Becoming Agile - What's involved?

GOAL	FUNDAMENTALS	CEREMONIES	TECHNIQUES
Deliver high-value, high quality working software quickly and frequently	*Short feedback loops* *Inspect and adapt* *Collaboration and teamwork*	*Training* *Project kick-off* *Release planning* *Iteration planning* *Daily stand-up* *Review / demo* *Retrospective*	Iterations User stories Co-location Task cards Storyboard Burn chart Grooming the backlog TDD Planning poker Done-done-done Pairing **Unit tests** Participate

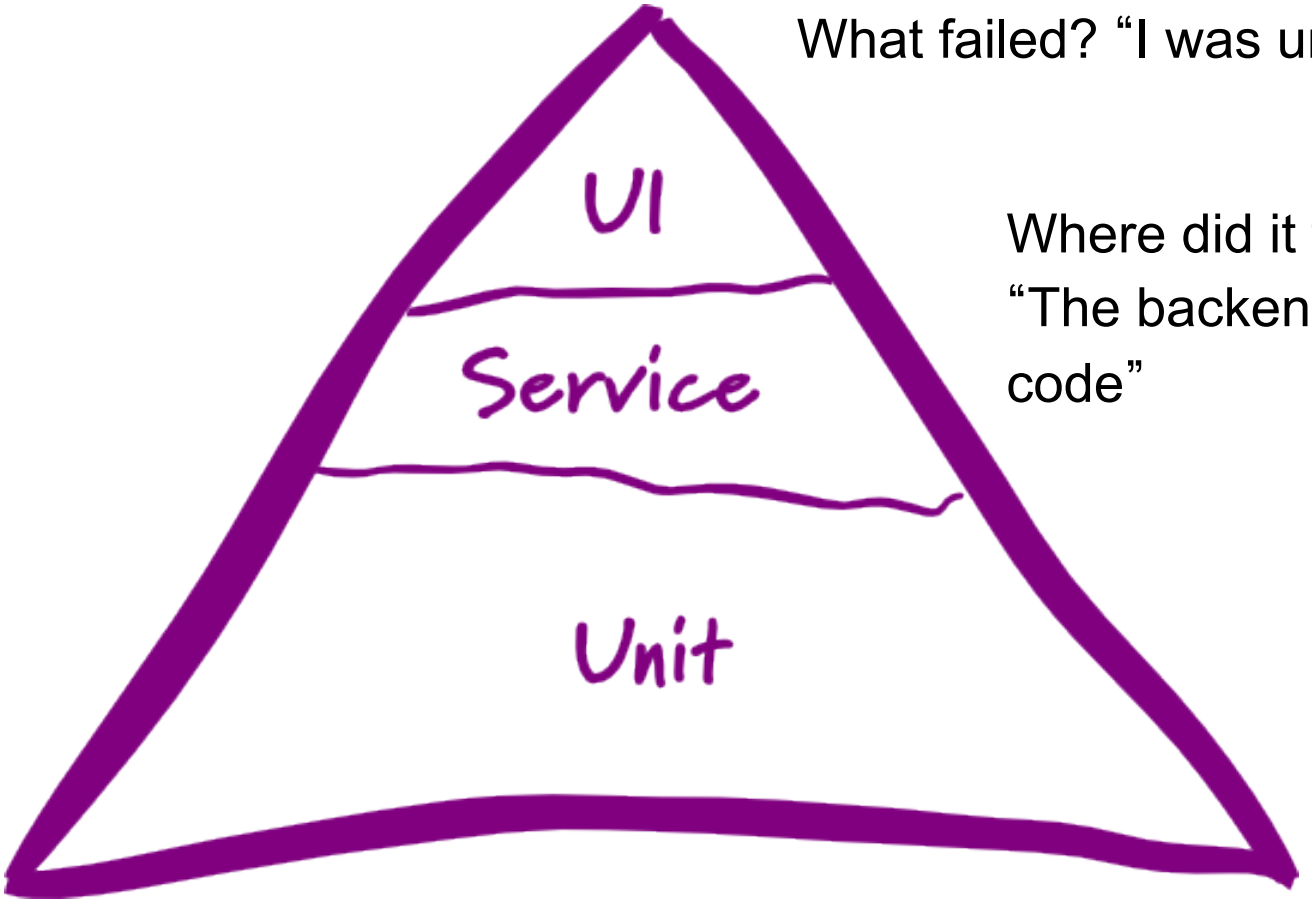
WORKING BETTER WITH SPECIFICATIONS - THE GIST

- Diagrams for communicating ideas
- Describe Specifications as code
- Keep the specifications along side code
- Test Code against specifications
- People for exploratory testing

Let’s see how the above work

THE TEST PYRAMID

ThoughtWorks®



What failed? “I was unable to login”

Where did it fail?
“The backend service returned an error code”

Why did it fail?
“A specific parameter was missing”
(nothing to do with the password itself!)

http://martinfowler.com/bliki/TestPyramid.html

SPECIFICATIONS AS CODE – document behaviour

```
describe Order do
  describe "#submit" do

    before do
      @book = Book.new(:title => "RSpec Intro", :price => 20)
      @customer = Customer.new
      @order = Order.new(@customer, @book)

      @order.submit
    end
```

Rspec Tests

```
def test_making_order
  book = Book.new(:title => "RSpec Intro", :price => 20)
  customer = Customer.new
  order = Order.new(customer, book)

  order.submit

  assert(customer.orders.last == order)
  assert(customer.ordered_books.last == book)
  assert(order.complete?)
  assert(!order.shipped?)
end
```

Find movies
==========

We are creating a new website to find and book movie tickets!

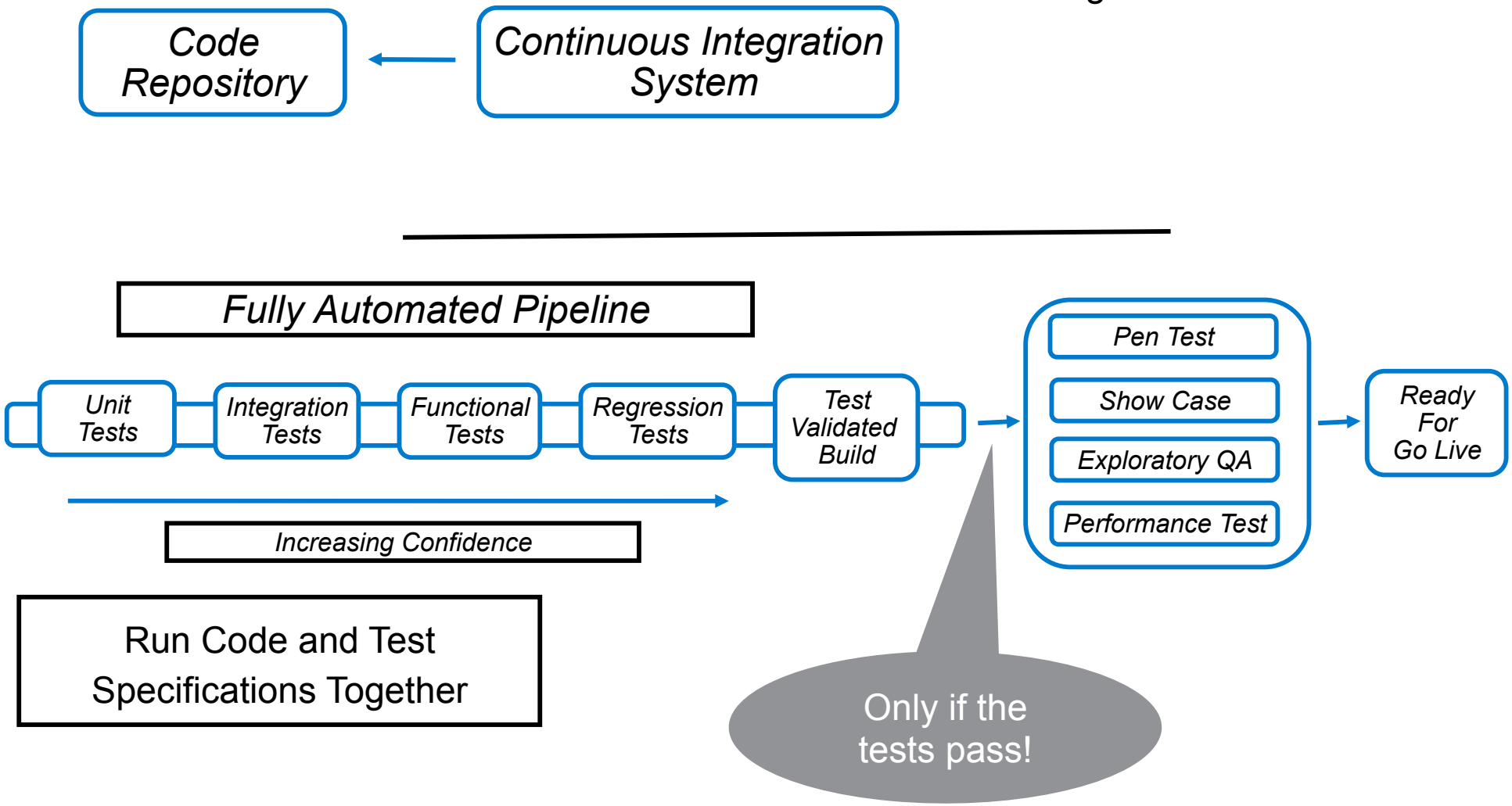
Location	*Movie*	*Theatre*	*Time*
Bangalore	Star wars	INOX	19.30
Mumbai	Star wars	INOX	14.20

Search for movies
---

* **Specify location as *<Location>***
* **Search for movie *<Movie>***
* **Verify that *<Theatre>* is playing *<Movie>* at *<Time>***

SPECIFICATIONS AS CODE – document behaviour

A software agent that frequently runs code and tests together



Requirements and Acceptance Criteria

- Store with code as much as possible

Benefits.

- Version Controlled along with code, not in a Network folder
- Usable by all, not just by Architects
- Kept up to date!
- Software is "Ready" only if the specifications are met
- Easy for new team members to ramp up – instant feedback if changes break something – helps with learning!

Questions to ask the IT team

- Are your critical business software documented via ”specification as code”?
- Do you have a roadmap to do so?
- What about infrastructure?
- If everything is documented, why are there still defects?
- How responsive are your off the shelf software vendors to change? (Defects, enhancements, changes in policy and business needs)

THANK YOU

For questions or suggestions
ram@thoughtworks.com
www.sriramnarayanan.com

ThoughtWorks®