

# A General Survey of the Modern QA Org

**Robb Nicholas**

Software Development Engineer in Test  
Apptio

# Outline

- Part 1: How to do QA
  - 1.1 How to test software
  - 1.2 How to make sure your app is ready for release (bleh, documentation)
- Part 2: How to bootstrap your own QA org
  - 2.1 How much testing does your product need
  - 2.2 Finding and Equipping people for success

# Why do you need QA

CCP: Eve Online  
Dec. 5<sup>th</sup>, 2007  
Boot.ini debacle



# Writing Tests

Or, how do you test the user experience?

## Manual Testing vs. Automated Testing

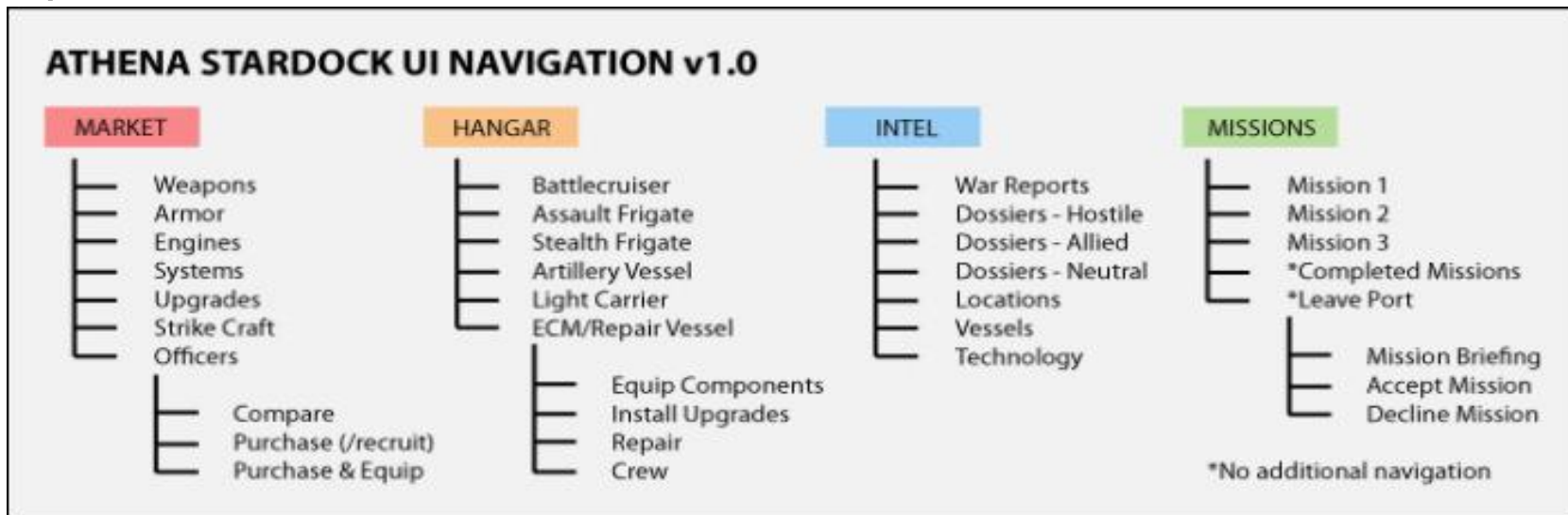
# Testing Functionality

Or, Pressing all the buttons

- Testing 101: If it says I press the player 1 Start button, then the game should start
- Map the UI to ensure completeness
- Execute Use Cases

# Mapping your UI

How do you know you've tested everything if you don't have a map?



# Use Cases

- Executing your use case (The Happy Path)
- Go through each decision point in your use case
- Generate test cases from each decision point.

# Testing Code

Enter The SDET

## The Prototypical API Testing Example

```
int squareNumber( int x )  
{  
    return x * x;  
}
```



# Testing Code

## Mapping variables

- Make a map of your variables, then reduce
- Only look at the interesting stuff
- Boundaries are interesting
- Zeroes are interesting
- Null pointers and Empty strings are interesting

# Test Types

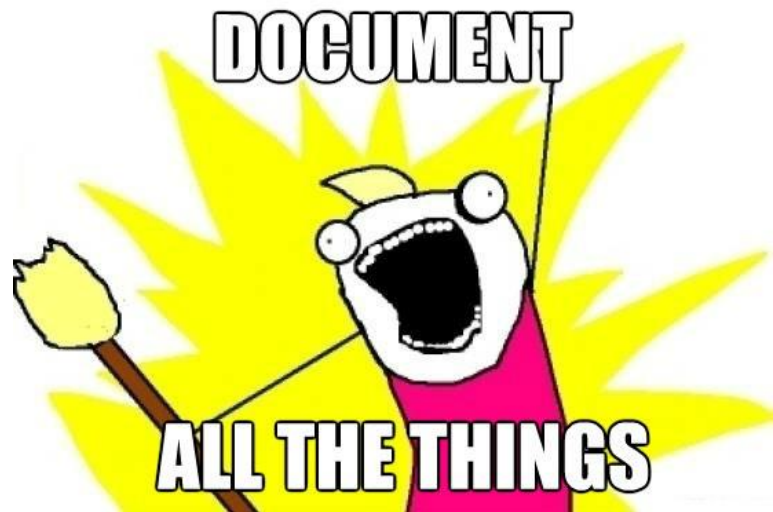
Test Type	Description
Happy Path	Verifying it works
Negative Tests	Verifying it fails
Integration Tests	Do your subsystems communicate correctly?
Regression Tests	Test your previous failures
End-To-End Tests	Can you play a complete game?
Boundary Tests	Exploring behavior

# Testing Algorithms

Things you should consider when testing algorithms.

- You'll need data, and lots of it
- That data needs Ground Truth Metadata
- Gathering data for your algorithm is a blast
- Living with lots of data is NOT a blast

# How to Tell Your Product is Ready



# The Test Plan

Describe how you plan on testing your application

Requirements:

- What are you going to test
- What are you NOT going to test
- How you're going to test the various test categories
- How you know your product is ready to ship

# The Test Specification

What are the actual tests you need to run

Requirements:

- Tests for each test category
- Priority – or how interested am I in the results
- What are you expecting to see

# Part II

## So you wanna make a Test org

# How big a test org do you need?

- How do you figure out how much testing you'll need?
- What kinds of people will you need?
- What are the tools they'll need to do their job?
- How do you even know what to do with them once you have them?



# How do you figure out how much testing you'll need

What are the manual tests you're expecting

What are the automated tests you're expecting

# What kinds of people do you need And how to hire them

QAE – Quality Assurance Engineer

SDET – Software Development Engineer in Test

# What tools might your testers need?

Utility Name	Description
Bug Tracker databases	Track bugs, find regressions, etc.
Test Harnesses	Provides test hooks to automate manual tests
Code Coverage	Shows what lines are being hit by test suites
Profilers	Identify performance bottlenecks

# Effective use of resources

Embedded vs. Team-Based

The Triage Meeting

The WAR Room

# Conclusion

Something Something Something  
Dark Slide