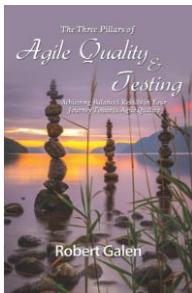


Agile Testing: *Techniques to Survive & Thrive*



Bob Galen
bob@rgalen.com

Mary Thorn
marythorn@gmail.com



1

1/2-day Workshop Outline

1. Introduction
2. Role of Testers
3. 3 Amigos and Story Writing
4. RBT & Test ideas
5. Mindmapping
6. Exploratory Testing
7. Wrap up



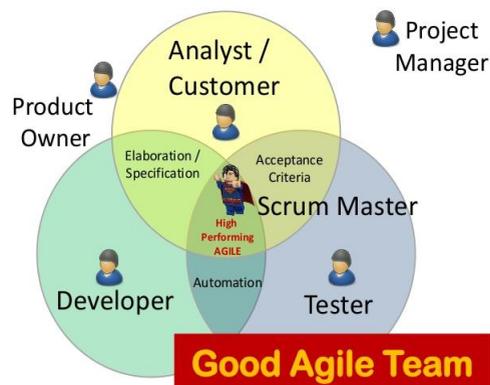
Role of Testers

- The testers foster a “Whole Team” view towards quality—focusing less on “Testing” and more on “Quality Practices & the Customer”
 - Serving as guides for the team; Testing the “hard bits”
 - Facilitating exploratory testing sessions—finding more interesting / valuable tests
 - Working with the Product Owners—are we solving the customers problems?

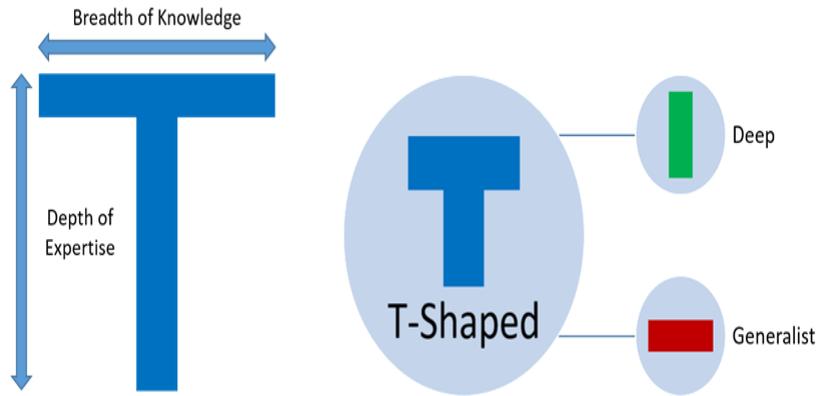


Role of Testers

- Customer Advocate
- Business Analyst
- Manual Tester
- Automation Tester
- Release Engineer
- Development Peer
- 1/3rd of the 3 Amigos
- Scrum Master
- Agile Process Champion
- Risk Raiser
- Project coordinator
- Etc...



T-Shaped



Copyright © 2018 RGCG, LLC

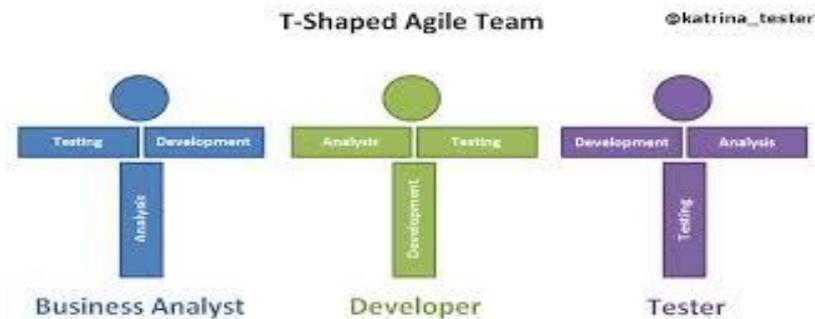


7

7

T-Shaped

- We're still working on the balance between roles and expectations, and the balance shifts, typically in response to the market.



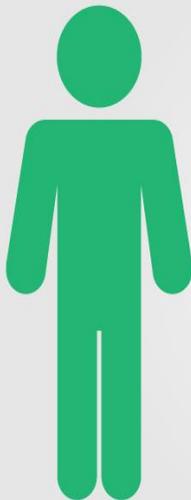
Copyright © 2018 RGCG, LLC



8

8

Characteristics of a T-shape tester



- ✓ They should have good communication skills.
- ✓ They should enjoy team work.
- ✓ They should be passionate and committed.
- ✓ They should develop empathy with other cultures.
- ✓ They should be creative.
- ✓ They should be action and quality oriented.

Copyright © 2018 RGCG, LLC



9

9

Self-organizing Agile Team

- Requisite skills to deliver a vertical slice of functionality
- Roles are defined
- Swarming around work
- Team based planning, sizing, decision-making
 - Who works on what
- Team has a determined capacity
- Team focuses on continuous improvement
- There is NO Scrummer-Fall

Copyright © 2018 RGCG, LLC



10

10

Retrospectives....



- There are many places to “make a difference”
 - Getting the 800 lb. Gorillas out on the table; Showing courage; telling truth
 - Fostering continuous improvement within the team
 - Setting the example; showing vulnerability—admitting you’re wrong
 - Team listening; active planning; dependencies; pairing
 - Risk-taking; Failure!

Guilds for X-Cutting Concerns

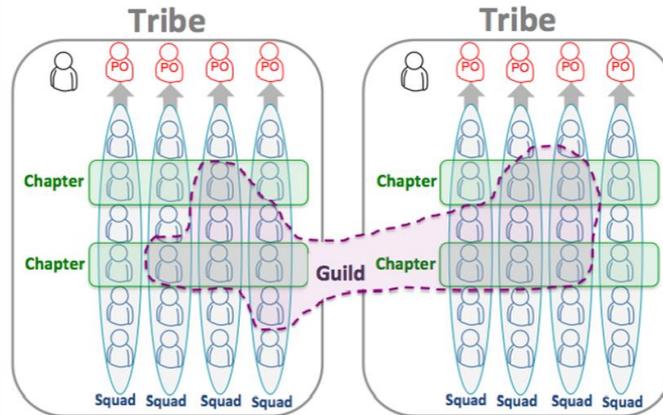
- Guilds are a self-organizing group of people with common interests. It is a natural forum for social interactions that build relationships that, in turn, promote cooperation, cohesion, and productivity.
 - Guilds provide a horizontal communication layer across teams.
- Engineers, testers, architects, U/X, and other staff use them to
 - Set their own missions, to establish technical roadmaps, to take on joint tasks for their grassroots initiatives, and to promote education through experiential learning.

Scaling Agile @ Spotify

with Tribes, Squads, Chapters & Guilds

Henrik Kniberg & Anders Ivarsson

Oct 2012



Copyright © 2018 RGCG, LLC



13

13

Do Agile Teams Have Constraints?

- Definition of Done
- Definition of Ready
- Meta-requirements
 - For example, at EMC: "Thou shall not loose data"
 - At Amazon: "Every purchase transaction needs to close"
- Embedding constraints into AC
 - Non-functional requirements
 - Collaboration, for example: Code reviews, test plan reviews, paired inspections
 - Event readiness: Demo planning
 - Environment promotion: Demo done from QA environment

Copyright © 2018 RGCG, LLC



14

14

Breakout: Discussion around the Agile Tester

- At your tables...
- First discuss the “perfect persona” of the agile tester. From what you heard, what do they “look like”?
- Given that persona
- Discuss the real-world gaps you’ve seen between that tester and your own (or others).
- And explore, HOW do we close the gaps?
- Be prepared to share some of your gaps/strategies...

User Stories

- Notion of Stories derived from Extreme Programming – XP
- Stories are requirements / features / units of work
- They are similar to use cases (as good a comparison as any)
- Written on a 3x5 card or Post-it note
- Capture a succinct item of work or functionality
- Are collaboratively defined by PO / Customer w/the team
- Are prioritized by and delivered to the PO / Customer

User Story Construct

As a <role>

I want <system behavior>

So that I realize <some business value>

**And can see that it does <example>

User Stories

- 3 Parts of a story, 3-C's by Ron Jeffries
 - Card
 - Conversation
 - Confirmation
- Cards are *intentionally vague or incomplete* in order to foster conversation and an emergent solution
- Confirmation (Acceptance Tests) are the specific things required to consider the story 'done' or acceptable to our customer

High 15

As traveler I want to get directions between two points, so I can get to my destination

- Verify I can enter addresses in all 50 states
- Verify directions can cross state boundaries
- Verify that directions are with +/- 5% mileage of MapQuest
- Verify that addresses can be swapped for return
- Verify that invalid addresses are handled w/error message

Product Owner Role: Card : Conversation : Confirmation



- **Front of card** – describes the Story characteristics – flexible format
 - 1-2 sentences; short and succinct
 - I've seen limits of 5 words per Story
 - Could be task-oriented, a milestone, simple work, virtually anything the team needs to do
 - Don't get too hung up on 'phrasing'

As a dog owner, I want to sign-up for a kennel reservation over Christmas so that I get a confirmed spot

As the Google Map interface, setup a mechanism for frequent Travelers to save and share their maps, so that they save time

Product Owner Role: Card : Conversation : Confirmation



- Product Owner must participate in defining 'doneness' criteria for card(s)
 - Completed with heavy tester input and buy-in from all team members
 - Unlike Cards, confirmations are normally quite specific
 - Confirmations document what must be demonstrated to complete the story
 - Confirmations are synonymous with Acceptance Tests
 - Should try to automate them, perhaps using FitNesse or Cucumber?

Verify individual as a registered pet owner
Verify that preferred members get 15% discount on basic service
Verify that preferred members get 25% discount on extended services and reservation priority over other members
Verify that past Christmas customers get reservation priority
Verify that declines get email with discount coupon for future services

Product Owner Role: Card : **Conversation** : Confirmation

- Product Owner must be available for the emergent solution conversations
- Critical point of influence in the design / construction phase
- Conversations as a team are the most important component of the '3Cs'
- Should also consider actual customers or other customer proxies in the process as appropriate



Acceptance Tests

- Story: Customer withdraws cash
 - **As a** customer,
 - **I want** to withdraw cash from an ATM,
 - **so that** I don't have to wait in line at the bank.
- Acceptance Tests:
 - ✓ Verify that customer authentication works
 - ✓ Verify that the customer is limited to 3 transactions in an ATM session
 - ✓ Verify that sufficient balance is in place to support the transaction
 - ✓ Verify that overall transaction workflow take no longer than 5 minutes
 - ✓ Verify that the transaction is immediately viewable in the customers online access

*Do the acceptance test spur questions or conversations?
Examples?*

Behavior Driven Development a Potential Substitute for Story 'phrasing'

- **Given** - some initial context
 - **When** - an event occurs
 - **Then** - ensure some outcomes
 - Story: Customer withdraws cash
 - **As a** customer,
 - **I want** to withdraw cash from an ATM,
 - **so that** I don't have to wait in line at the bank.
 - BDD Scenario 1: Account is in credit
 - **Given** the account is in credit
 - **And** the card is valid
 - **And** the dispenser contains cash
 - **When** the customer requests cash
 - **Then** ensure the account is debited
 - **And** ensure cash is dispensed
 - **And** ensure the card is returned
- <http://dannorth.net/introducing-bdd>

Copyright © 2018 RGCG, LLC



31

31

Behavior Driven Development a Potential Substitute for Story 'phrasing'

- Story: Customer withdraws cash
 - **As a** customer,
 - **I want** to withdraw cash from an ATM,
 - **so that** I don't have to wait in line at the bank.
- BDD Scenario 2: Account is overdrawn past overdraft limit
 - **Given** the account is overdrawn
 - **And** the card is valid
 - **When** the customer requests cash
 - **Then** ensure a rejection message is displayed
 - **And** ensure cash is not dispensed
 - **And** ensure the card is returned

Copyright © 2018 RGCG, LLC



32

32

Story Attributes: INVEST



1. Independent ➤ as much as possible stories need to stand alone, with few dependencies to other stories
2. Negotiable ➤ they are not fixed “contracts”
3. Valuable ➤ prioritized so that the value to the customer is clear (relative to other stories)
4. Estimate-able ➤ clear enough, with knowledge surrounding domain & technical points
5. Small ➤ fit easily within an iteration, requiring 1 to a few resources to implement
6. Testable ➤ usually expressed with acceptance criteria and/or automated tests

Copyright © 2018 RGCG, LLC



33

33

User Story Research Story or Research Spike



- A Research Spike is a story focused toward technical (or feasibility) investigation
 - ❑ Only exception for “Working Code”; result is information or knowledge; not production code
 - ❑ Focused on Complex and/or Risky User Stories
 - ❑ Typically leads to other development-centric or (Ready) User Stories and thoughtful estimate(s)
 - ❑ Can plan for “swarming on” the resulting Stories
 - ❑ Written as a story with specific Acceptance Criteria
 - ❑ I like to time-box their execution (not estimate); Ex: Bob and Carol for 2 days.

Copyright © 2018 RGCG, LLC



34

34

3-Amigos



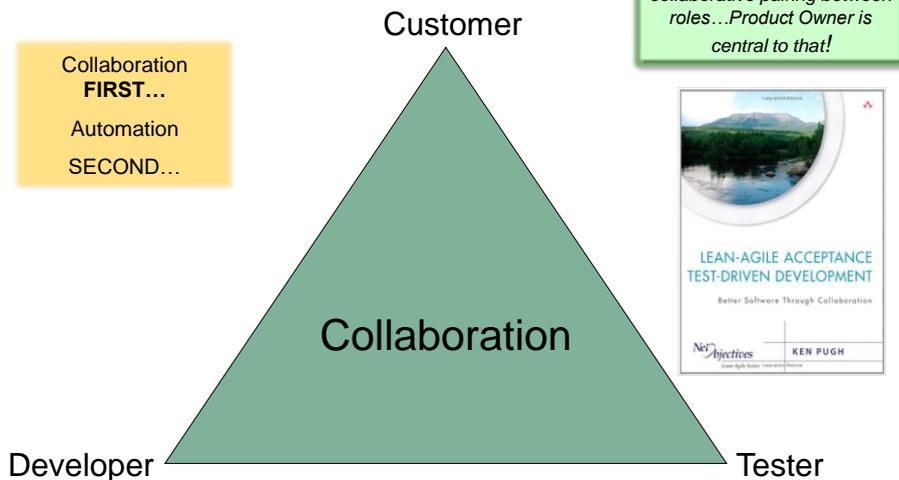
- Coined by George Dinwiddie
 - <http://rgalen.com/agile-training-news/2014/4/13/3-amigos-in-agile-teams>
- Swarming around the User Story by:
 - Developer(s)
 - Tester(s)
 - Product Owner
- Conversation device – reminder for collaboration amongst relevant team members

3-Amigos



- Often takes the form of a:
 - Feature team
 - Story owner team
- Surrounding the life-cycle of a story from:
 - Story inception
 - Epic writing
 - Via Backlog Refinement:
 - Iterative - Feature/Theme decomposition (estimation-driven)
 - Iterative - Story emergence – Ready?

Remember -- The Triad



Copyright © 2018 RGCG, LLC



38

38

Breakout: Writing Stories for MS Word

- Get into groups of 2-3 collaborators
- Write 5-10 user stories focused on High-level functionality of MS Word
 - (or another feature reach application)
- Use (As a..., I want..., So that...) format
- Each story should have ~5 Acceptance Criteria in (Verify that...) format
 - Functional and Non-functional
- Be prepared to share a few...

Copyright © 2018 RGCG, LLC



39

39

Risk Based Testing

How to Manage Risks During Test Planning



Copyright © 2018 RGCG, LLC



40

40

Risk-Based Testing Background

- It starts with the realization that you can't test everything – ever!
100% coverage being a long held myth in software development
- There are essentially 5 steps in most of the models
 1. Decompose the application under test into areas of focus
 2. Analyze the risk associated with individual areas – technical, quality, business, schedule
 3. Assign a risk level to each component
 4. Plan test execution, based on your SDLC, to maximize risk coverage
 5. Reassess risk at the end of each testing cycle

Copyright © 2018 RGCG, LLC



41

41

Risk-Based Testing Background

- Risk-Based Testing is effectively a risk mitigation technique
 - Not a prevention technique
- It's about trade-offs
 - Human and physical resources
 - Ratio's between Producers (Developers) and Consumers (Testers)
 - Time
 - Rework (retesting & verification)
 - Quality – Coverage vs. Delivery
 - Visibility into the trade-offs

Copyright © 2018 RGCG, LLC



42

42

Test Ideas

- What are they?
 - Risked based test planning technique
 - Created by Rob Sabourin
 - Replaces traditional waterfall test plan in Agile.



Copyright © 2018 RGCG, LLC



43

43

Test Ideas

Identifier	Focus	Test Objective	Business Importance	Technical Risk	Priority
TID0010	Capabilities	Produce correct box of chocolates based on manifest	LOW	SIGNIFICANT	5
TID0100	Failure Modes	What if it runs out of paper	LOW	SIGNIFICANT	5
TID0170	Usage Scenarios	Can operator stop system	LOW	SIGNIFICANT	5
TID0260	Outcome	Can we produce correct daily reports	LOW	SIGNIFICANT	5
TID0020	Failure Modes	Are there gaps in a box	MEDIUM	SIGNIFICANT	4
TID0040	Capabilities	Can it fill boxes with mixed chocolates	LOW	NEUTRAL	4
TID0110	Failure Modes	What if it runs out of other supplies	MEDIUM	SIGNIFICANT	4
TID0140	Failure Modes	What if operator enters incorrect data in manifest	LOW	NEUTRAL	4
TID0180	Usage Scenarios	Can emergency repairs be done without stopping production	MEDIUM	SIGNIFICANT	4
TID0200	Usage Scenarios	Can production be resumed after emergency repairs	LOW	NEUTRAL	4
TID0270	Outcome	Can we produce correct monthly reports	MEDIUM	SIGNIFICANT	4
TID0290	Input	Can we vary boxes with different speeds of conveyors	LOW	NEUTRAL	4
TID0030	Capabilities	Can it wrap chocolates with ribbons	LOW	SIGNIFICANT	3
TID0050	Capabilities	Can it fill boxes with one type of chocolates	MEDIUM	NEUTRAL	3
TID0070	Input	Vary Combinations of Ribbons, Paper, Boxes	LOW	MINIMAL	3
TID0120	Failure Modes	What if machine drops chocolate but continues to try wrapping (in process)	LOW	SIGNIFICANT	3
TID0130	Failure Modes	What if operator enters WRONG manifest	MEDIUM	NEUTRAL	3
TID0150	Failure Modes	What if something else in conveyor belt not chocolate	MEDIUM	NEUTRAL	3
TID0160	Quality Factors	Is system easy to stop	LOW	SIGNIFICANT	3
TID0210	Usage Scenarios	Can loader load supplies	MEDIUM	NEUTRAL	3
TID0230	Usage Scenarios	Can loader add ribbons while production is in progress	LOW	MINIMAL	3
TID0250	Outcome	Can we produce correct batch report	LOW	SIGNIFICANT	3
TID0300	Input	Can we have batches with high percentage of one type of chocolate	MEDIUM	NEUTRAL	3
TID0060	Capabilities	Can we support different sizes of chocolates in the same box	LOW	NEUTRAL	2
TID0080	Failure Modes	Mechanical failure does it handle it gracefully	MEDIUM	MINIMAL	2
TID0190	Usage Scenarios	Can emergency repairs be done stopping production	LOW	NEUTRAL	2
TID0240	Outcome	Produce correct reports	MEDIUM	MINIMAL	2
TID0280	Failure Modes	Will system ever run hot enough to melt the chocolate	LOW	NEUTRAL	2

Copyright © 2018 RGCG, LLC



44

44

Test Ideas - Sources

- Capabilities
- Failure Modes
- Quality Factors
- Usage Scenarios
- Creative Ideas
- States
- Data
- Environments
- White Box
- Taxonomies



Copyright © 2018 RGCG, LLC



45

45

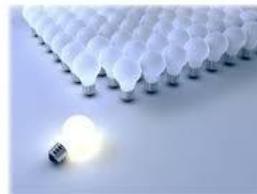
Test Ideas

- How to find them?
 - Does system do what it is suppose to do?
 - Does the system do things it is not supposed to?
 - How can the system break?
 - How does the system react to it's environment?
 - What characteristics must the system have?
 - Why have similar systems failed?
 - How have previous projects failed?

Test Ideas - Process

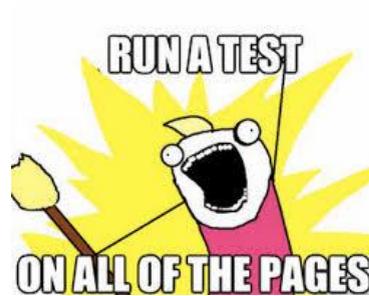
- Life of a test idea
 - Comes into existence
 - Clarified
 - Prioritized
 - Test Now (before further testing)
 - Test before shipping
 - Nice to have
 - May be of interest in some future relea
 - Not of interest in current form
 - Will never be of interest
 - Integrate into a testing objective

Test Your Big Idea



Test Ideas – 3 Amigos

- Test Triage Meeting
 - Review Context
 - Business – with PO
 - Technical – With Developer
 - Add or remove tests
 - Agree to where the cut line is



Breakout – Test ideas for Google.com

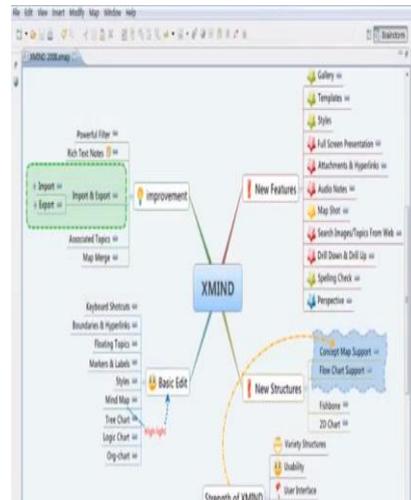


Google Search

I'm Feeling Lucky

MindMaps

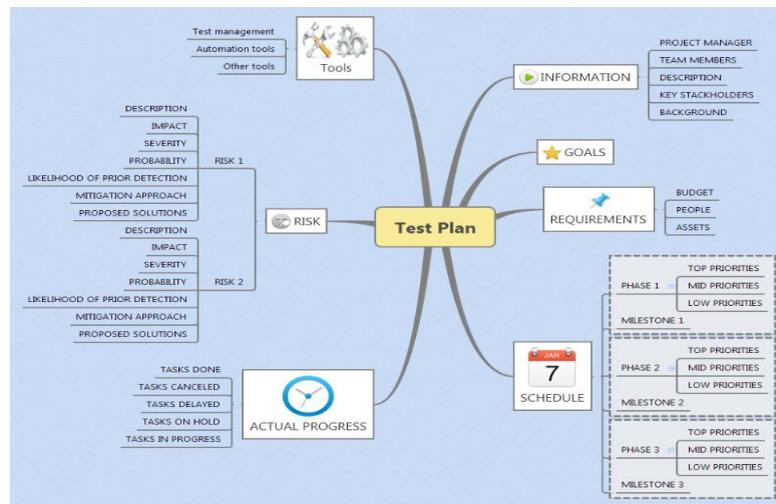
- Mindmaps can be used for:
 - ❑ **Test Planning**
 - ❑ Requirement analysis
 - ❑ Impact analysis
 - ❑ Task allocation
 - ❑ **Test case design**
 - ❑ **Traceability**
 - ❑ Test reporting -Quick test reports



MindMaps – Test Planning

- **Test planning:**
 - ❑ Draw an initial mindmap keeping in mind the list of tasks, schedules, tools, roles, responsibilities, milestones etc.
 - ❑ Present the mindmap and discuss it with your stakeholders.
 - ❑ Modify the mindmap if any changes are required.
 - ❑ All you might have to do is to add or remove a node/branch.
- The final mind map shows you the scope of testing in one glance.
- This mindmap can be used as a blueprint and later converted into a plan. This ensures that no test activity is missed.

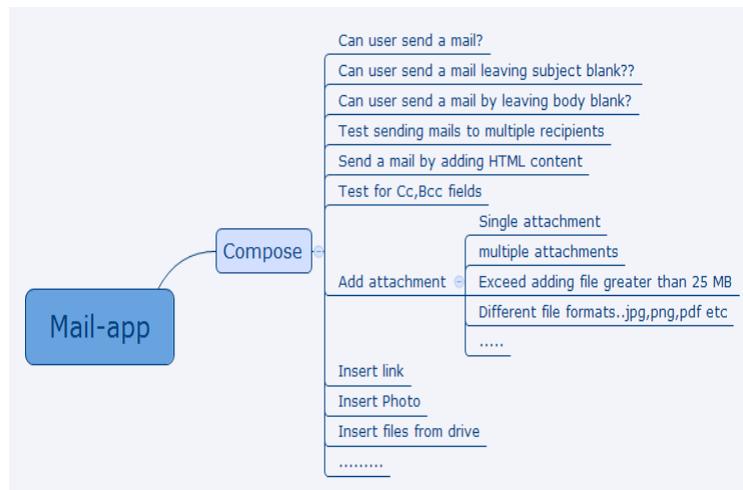
Breakout -MindMaps – Test Planning



MindMaps – Test Case Design

- **Test Case Design:**
 - An efficient way of creating lean test cases.
 - It reduces the time required for creating test cases yielding better results.
 - Easy to maintain and are flexible to changing requirements.
- Draw branches from every user story/epic and associate all its functionalities as sub-nodes.
- Start adding test case for each functionality.

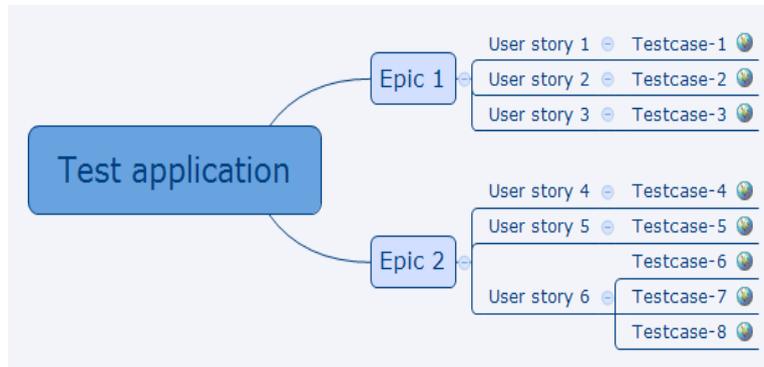
Breakout -MindMaps – Test Case Design



MindMaps – Traceability Mindmap

- A traceability matrix is
 - An essential tool for every tester to analyze and improve the test coverage. You can use a mindmap instead of a tabular traceability matrix.
 - Ensures that you have not missed out writing test cases for any user story
 - Gives you the birds-eye view of your test coverage. You can identify the areas where you need to strengthen your coverage.
- To create a traceability mindmap
 - Add nodes of all the Epics
 - Draw branches from every module and associate all its user stories as subsequent nodes.
 - Now link the test cases for every functionality. You can link the requirement number of the test management tool.

MindMaps – Traceability Mindmap



MindMaps – Tools

- There are tons of commercial and open-source tools that lets you visualize your ideas as a mindmap. I prefer the following tools:
 - Xmind(Windows/Mac/Linux)- Probably the most popular and free mindmapping tool.
 - Coogole(Web app) - Coogole is a web app that lets your team collaborate and work on a single mindmap.
- The use of mindmaps are getting popular with agile testers and lean test practitioners..

Breakout - MindMaps

- Let's take a look at www.coggle.com
 - We'll be "driving", but we expect you to login in later and try things out...
- Review:
 - Implementation of google.com

Copyright © 2018 RGCG, LLC



60

60

Exploratory Testing

- The technique was coined by Cem Kaner in his book
 - *Testing Computer Software*
- James Bach (www.satisfice.com)
 - considered the *father* Exploratory Testing in Practice
- His brother Jon Bach
 - has focused on managing ET focused projects -
- added Session Based extensions
- Often confused with ad-hoc testing.
Contrasting characteristics include:
 - Trainability, repeatability, focused or tasked, agile uses, and experience centered



Copyright © 2018 RGCG, LLC



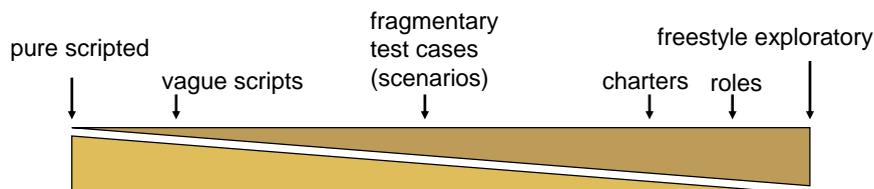
72

72

Exploratory Testing Team-Centric

- Amplifies the domain experience of your team members
 - People, skill and experience-centric
- Great technique for cross-training
 - Pairing individuals to leverage knowledge sharing
- Align with your most familiar testers (and developers)
 - You're counting on the product domain experience and testing skill to achieve success
 - Risk knowledge; quality practice knowledge

Exploratory Testing Scripted vs. Exploratory Continuum



To know where a test falls on this scale, ask yourself: *“to what extent am I in control of the test, and from where did the idea originate?”*

Exploratory Testing Analogies

- Job Interview
- Warfare Operations
- Newspaper Reporter
- Detective
- Others?
- Bounty Hunter
- 20 Questions
- Psychologist
- Going to a conference
- Lewis & Clark

Exploratory Testing Session Strategy

- Sessions
 - Exploratory Testing proceeds in a series of interconnected sessions that are focused on a specific testing project (application)
- Time-Boxed Charters
 - Planning the project encompasses establishing a set of time boxed session charters
- Roles
 - Establishing roles and focus areas for the sessions or groups of sessions
- Tempo
 - Establishing the session execution dynamics
 - Starting, Stopping, Re-Chartering, Reporting
- Reporting
 - Reporting progress to stakeholders & re-establishing the overall test strategy / charter

Exploratory Testing Session Dynamics

- Sessions are focused ET events
 - They are limited in duration (60-120 minutes)
 - They have a session charter, goal, or focus
 - The results of each session are captured in a log
 - Testing path logged, findings & bugs reported, repeatable steps
 - Sessions are de-briefed (retrospective) with re-chartering as required for subsequent sessions
 - A day of testing is composed of multiple sessions
 - Flip charts, charters on 4x6 cards, shouting out findings

Exploratory Testing Session Dynamics

- Often there is a sense of collaboration in the sessions
 - Paired testers; Paired w/developers or others
 - All-hands on deck...testing!
 - Co-located in the same room; lab area, conference room, war room, open space, etc.
 - Often look to share common data & testing environments
 - Setup of the environment is a quality step
 - Larger sessions usually need facilitators or “conductors”
 - Very often Sr. Testers fill this role
 - Everyone overhears activity & progress

Exploratory Testing Roles & Feature Areas

- **Roles, as a...**
 - Power user
 - Specific clients / configurations
 - User communities
 - Administrator
 - Tester, testability
 - Compliance user
 - Process owner
 - Business user
 - Negative tester – edge cases

- **Feature Areas of focus...**
 - Installation
 - Compatibility
 - Database integrity
 - 3'rd party add-ins
 - Configuration & setup
 - Usability
 - Performance & Load
 - New Features
 - Online help and docs
 - Security
 - Interoperability
 - Beta

Exploratory Testing What to Write While Exploring...

Observations <i>(To the degree you think they are relevant to stakeholders)</i>	<ul style="list-style-type: none"> • feature model • text from log files • text from dialogs • bugs
Conjectures <i>(Inferences based on experiences)</i>	<ul style="list-style-type: none"> • test ideas • questions • product and project issues • concerns • risks
Project information <i>(Independent of observer)</i>	<ul style="list-style-type: none"> • charter • test actions • config info • build details • tools used

Exploratory Testing

What to Write While Exploring...

- In other words, *breadcrumbs*
 - Where you've been, how you got there, what did you find?
 - Sufficient documentation to write test cases

- Should lead to other “test ideas”
 - Very often Exploratory Testing sessions drive other Charters / Sessions
 - Dynamic Coverage based on discovery
- Capture bugs quickly; enter them later

- *Prime Directive* – Cover as much bounded ground as possible!

Exploratory Testing

Sweet Spots

- Any extremely time constrained testing situation
- Anytime you have a lot of ambiguity (stability, feature operation, etc.)
- When you're blessed with lots of solid domain experience, SME breadth
 - Smoke testing; Does it work?
 - Acceptance testing; Did I get what I expected?
 - Beta testing; Will we embarrass ourselves?

- Agile testing – daily explorations: What works? What doesn't? Progress? Feedback for development...

Paired SBET Agile Benefits

- The impact of inviting customer support (and other x-functional team members) was HUGE
 - In understanding customer usage
 - In adding / changing Charters and focusing domain knowledge
- Developers increased their understanding & empathy for testing
 - It's Hard!
 - It's Long!
 - We need to leverage our unit test investments
 - It needed broader automation!

Exploratory Testing Supporting Tools

- Philosophically Exploratory Testing is not a tool based activity, it's a human experience based one. So tool requirements are minimized. That being said...
- The following can be useful –
 - Any tools that allow you to capture screen state information – ex: Spector
 - Quick, UI interaction tools – ex: Perlclip
 - Fast logging / scripting tools; ex: Log-Watch
 - Web based DTS
 - Wiki's, Notepad
 - TestExplorer

Breakout - Exploratory Testing

- Using either Google Maps, Google Drive, or Microsoft Word...
- Gather in groups of 4-6
- Strategize charters for the target application
 - Charter Name, Charter Focus or Boundaries
 - Boundaries can be: Test and !Test
 - Session length is 90 minutes, so use that for thoughtful decomposition
- Prioritize charters into High, Med, Low impact for general usage
- Be prepared to discuss (5 min readout) of your work

Wrapping up...

- There are a lot of old and new testing techniques that can be used to enhance your agile testing journey.
- Here we discussed just a few...
- Read blogs, go to conferences, read our book 😊



Contact Info

Bob Galen

Principal Consultant,
RGalen Consulting Group, L.L.C.

***Experience-driven agile focused training,
coaching & consulting***

Cell: (919) 272-0719

bob@rgalen.com www.rgalen.com

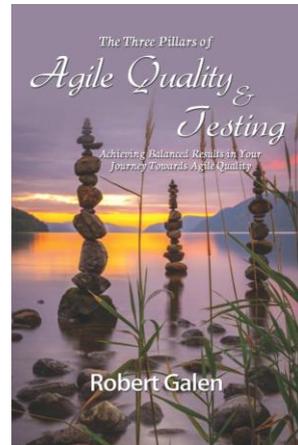
bgalen@velocitypartners.net www.velocitypartners.net

Blogs

Project Times - <http://www.projecttimes.com/robert-galen/>

BA Times - <http://www.batimes.com/robert-galen/>

Podcast on all things 'agile' - <http://www.meta-cast.com/>



Copyright © 2018 RGCG, LLC



112