thoughtworks

Vapasi

Role of a QA

Goal

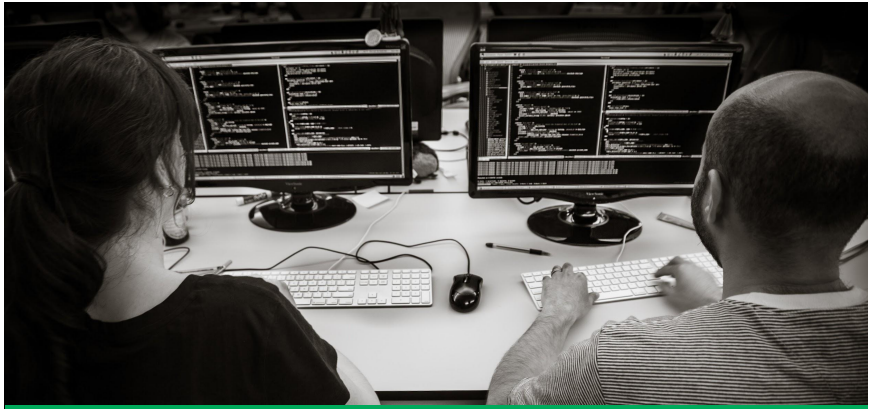
- What is a QA ?
- how does that fit in with other roles?
- What hats do QAs wear on delivery teams ?

WHAT DOES QUALITY MEAN?

WHAT DOES QUALITY MEAN?

BUILD THE *RIGHT* PRODUCT

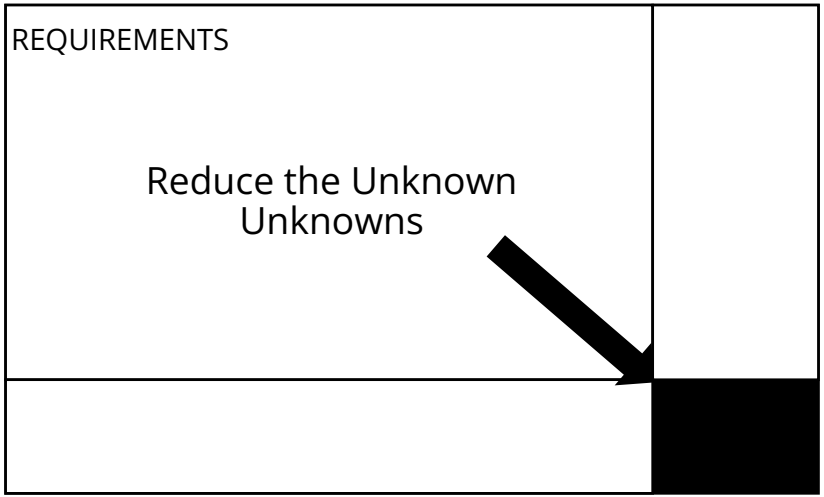
- What's important for the Customer
  - *Said vs Unsaid Expectations*
- Alignment with Business Priorities



BUILD THE *PRODUCT* RIGHT

- Quality Engineering
  - *Build in Quality vs Only Test for Quality*
  - *Fast Feedback*
  - *CICD Process*
  - *Tools & Frameworks*

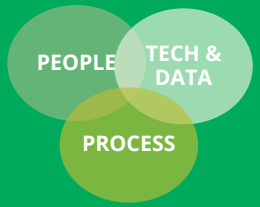
BUILDING THE *RIGHT* PRODUCT



INCEPTIONS: REQUIREMENTS ELICITATION

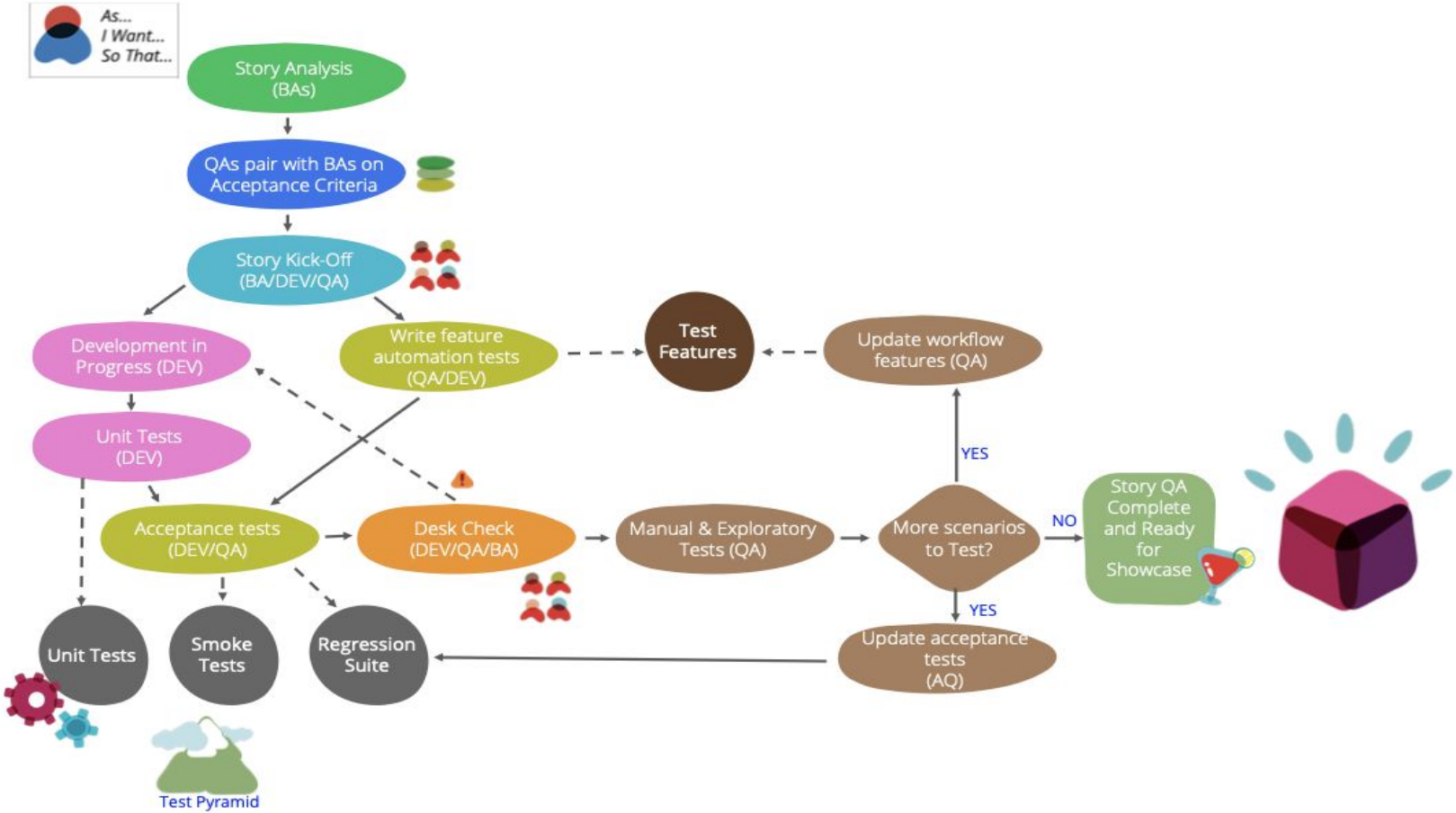
Project vision, goals and scope | High level processes & workings | Key risks, issues, and delivery constraints | x-Functional Implementation & Testing Approach

CROSS FUNCTIONAL REQUIREMENTS



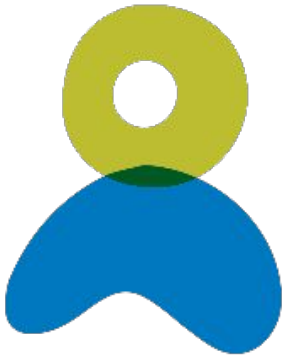
ACCESSIBILITY	ARCHIVABILITY	AUDITABILITY	AVAILABILITY	COMPATIBILITY	CONFIGURABILITY	CONTINUITY	DATA INTEGRITY
EXTENSIBILITY	INTEROPERABILITY	LEGAL	LOCALISATION	MAINTAINABILITY	MULTI-ENV SUPPORT	PERFORMANCE	PERSONALISATION
PRIVACY	RELIABILITY	REPORTING	RECOVERABILITY	STABILITY	SECURITY	UPGRADEABILITY	USABILITY

BUILDING THE *PRODUCT* RIGHT



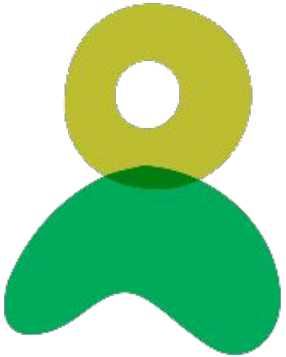
Who is responsible for Quality of project ?

What does QA stand for?



Quality assurer

Quality Advocate



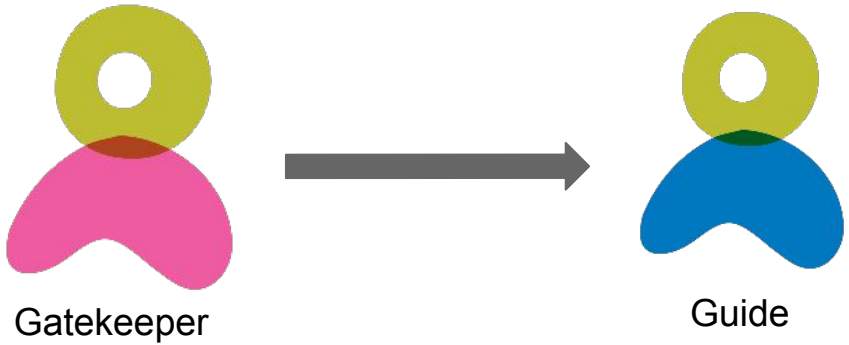
Quality Analyst

QA Primary Responsibility

Advocate for processes that ensure the product being developed meets the needs of...

- End Users
- The Organization

...now and into the future



Analogy shared JK Werner, BA/PM in SF

Meeting Users' Needs ?

Trying to find problems with an existing application

Who are our users?
How will they use this?
Are we meeting their need?

- Functionality is relevant
- System is reliable
- Performance is sufficient

Meeting Organizations' Needs ?

Test a product designed in a vacuum

Why is this product important?
What does the business want from it?
Does the application meet that need?

- Ability to respond to quickly changing business goals
- Money generating? Data generating?

Why to advocate ?

What practices to advocate ?

What to avoid ?

WHAT QUALITY SHOULD NOT BE ABOUT

0
Bug Count

100%
Automation

100%
Onus on QA

0
Tech Debt

How does QA fit in Agile team?

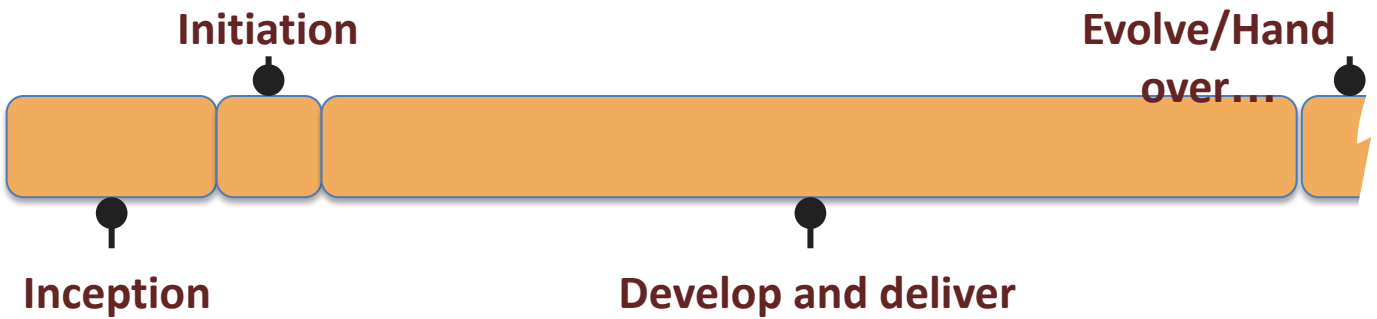
Agile Manifesto

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

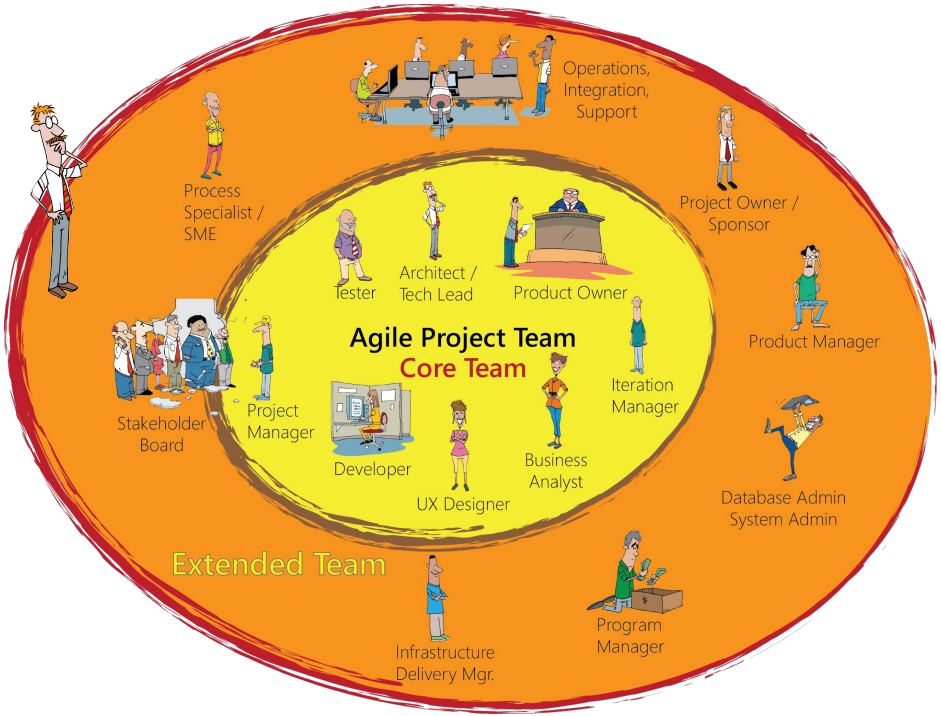
Agile Testing Manifesto

- Collaborative ownership OVER detached objectivity
- Targeted automation OVER widespread anti-regression
- Defect prevention OVER defect reporting
- Exploratory testing OVER predetermined scripting

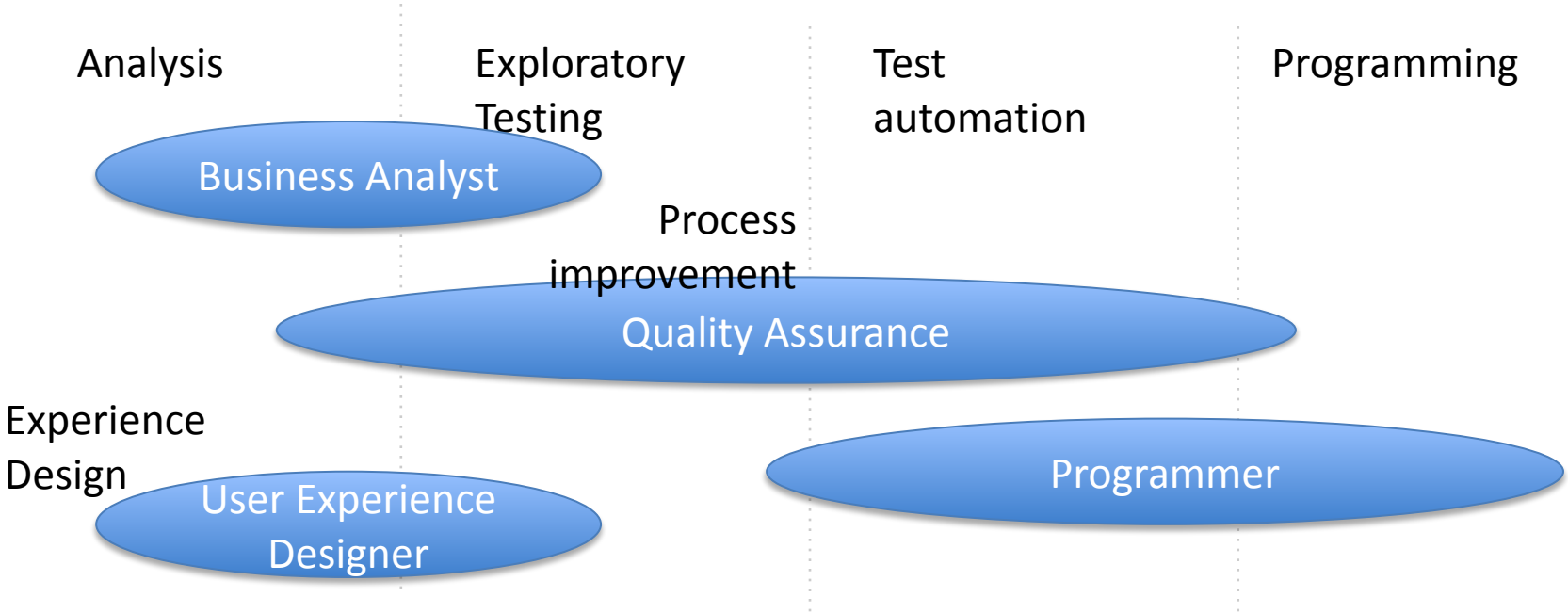
Agile Project Lifecycle

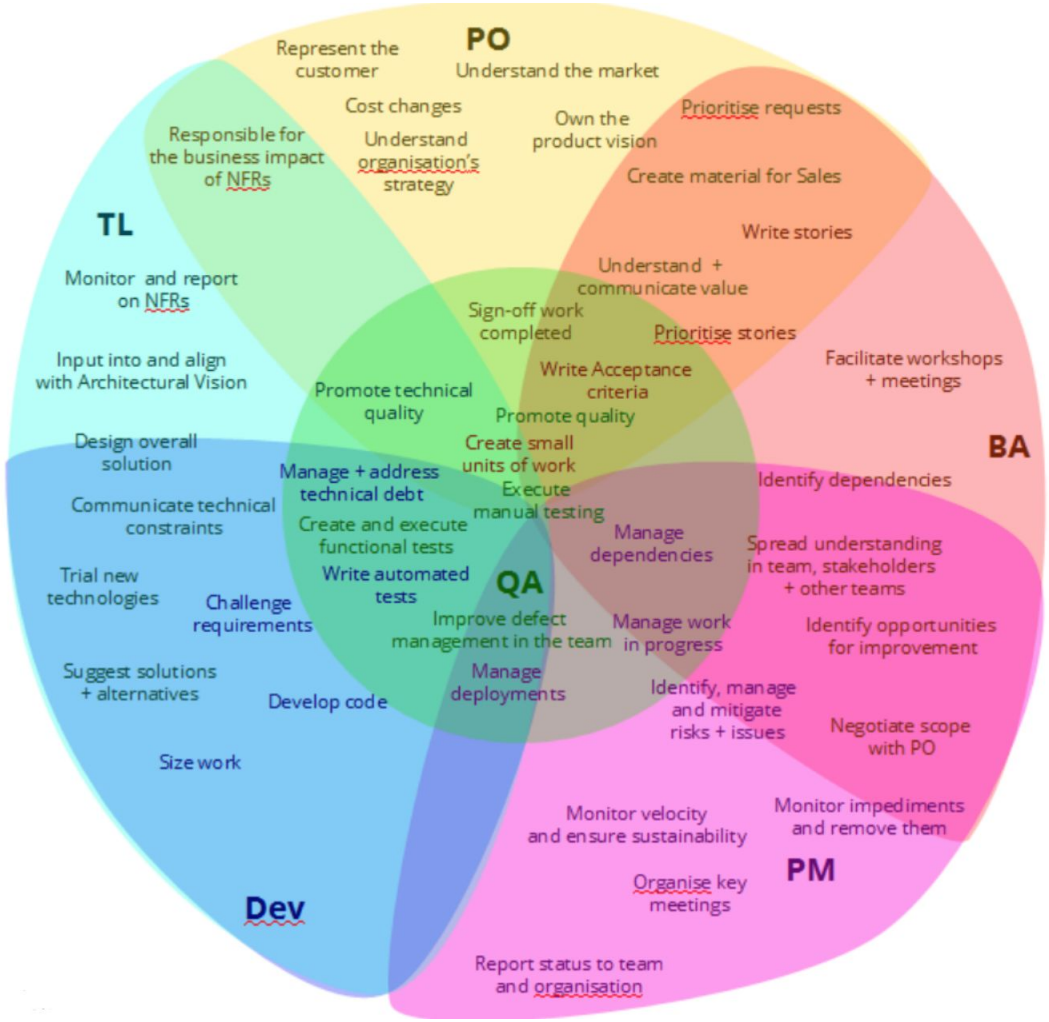


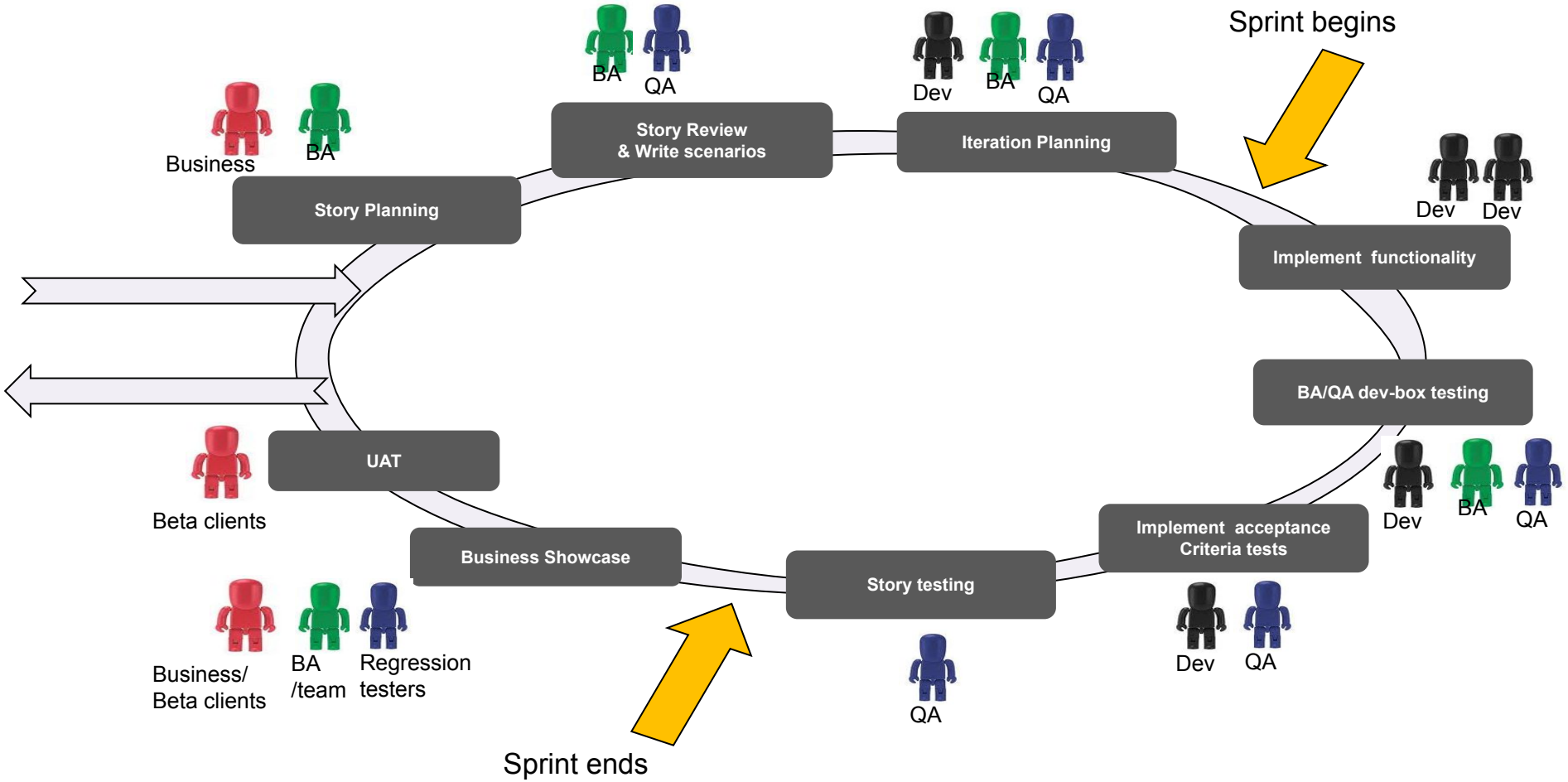
Team Composition



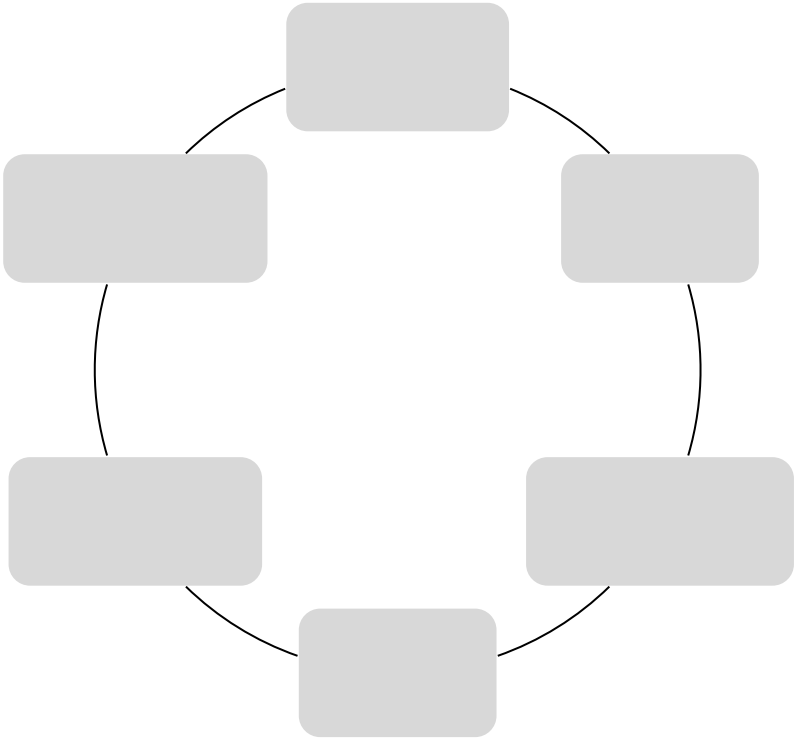
Team Activities by Role

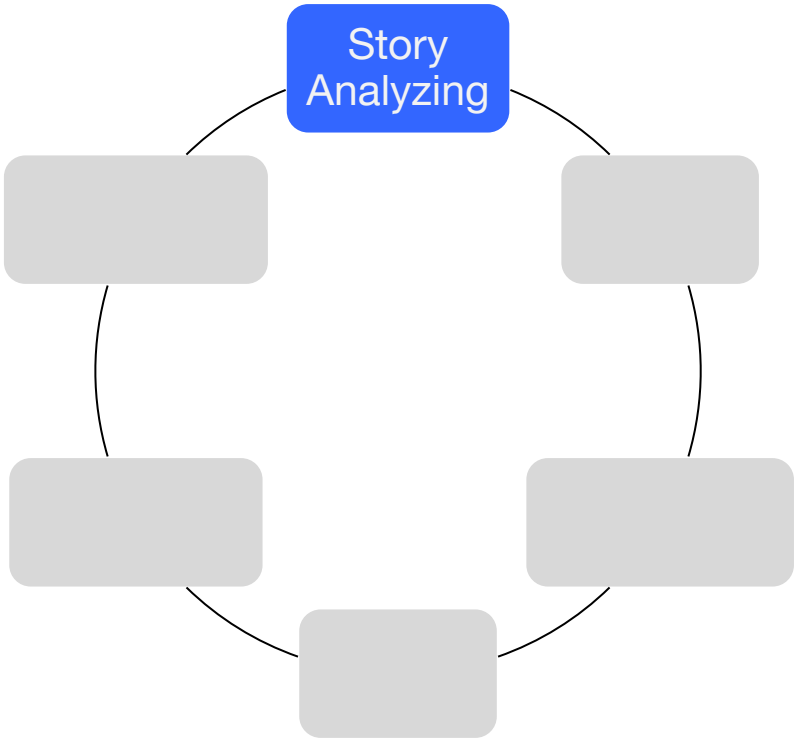



PO
Represent the customer
Understand the market
Cost changes
Own the product vision
Prioritise requests
Responsible for the business impact of NFRs
Understand organisation's strategy
Create material for Sales
TL
Write stories
Understand + communicate value
Monitor and report on NFRs
Sign-off work completed
Prioritise stories
Input into and align with Architectural Vision
Write Acceptance criteria
Facilitate workshops + meetings
Promote technical quality
Promote quality
Design overall solution
Create small units of work
BA
Manage + address technical debt
Identify dependencies
Execute manual testing
Communicate technical constraints
Create and execute functional tests
Manage dependencies
Spread understanding in team, stakeholders + other teams
Trial new technologies
Write automated tests
QA
Challenge requirements
Improve defect management in the team
Manage work in progress
Identify opportunities for improvement
Suggest solutions + alternatives
Manage deployments
Identify, manage and mitigate risks + issues
Develop code
Negotiate scope with PO
Size work
Monitor velocity and ensure sustainability
Monitor impediments and remove them
PM
Organise key meetings
Dev
Report status to team and organisation

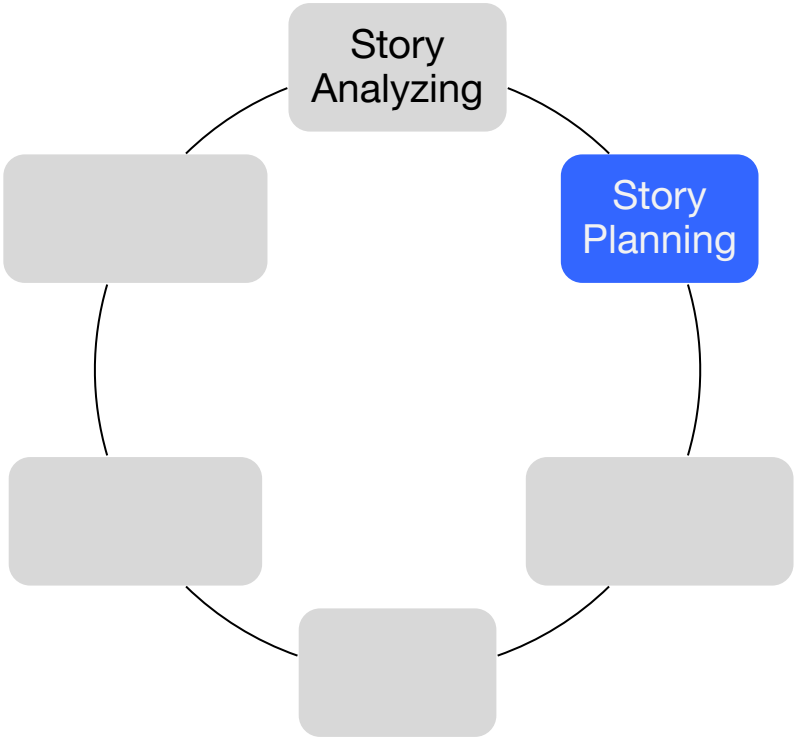

Sprint begins
BA
QA
Dev
BA
QA
Business
BA
Story Review
& Write scenarios
Iteration Planning
Story Planning
Dev
Dev
Implement functionality
BA/QA dev-box testing
Beta clients
UAT
Dev
BA
QA
Implement acceptance
Criteria tests
Business Showcase
Story testing
Business/
Beta clients
BA
/team
Regression
testers
Dev
QA
QA
Sprint ends

QA And The Story Lifecycle

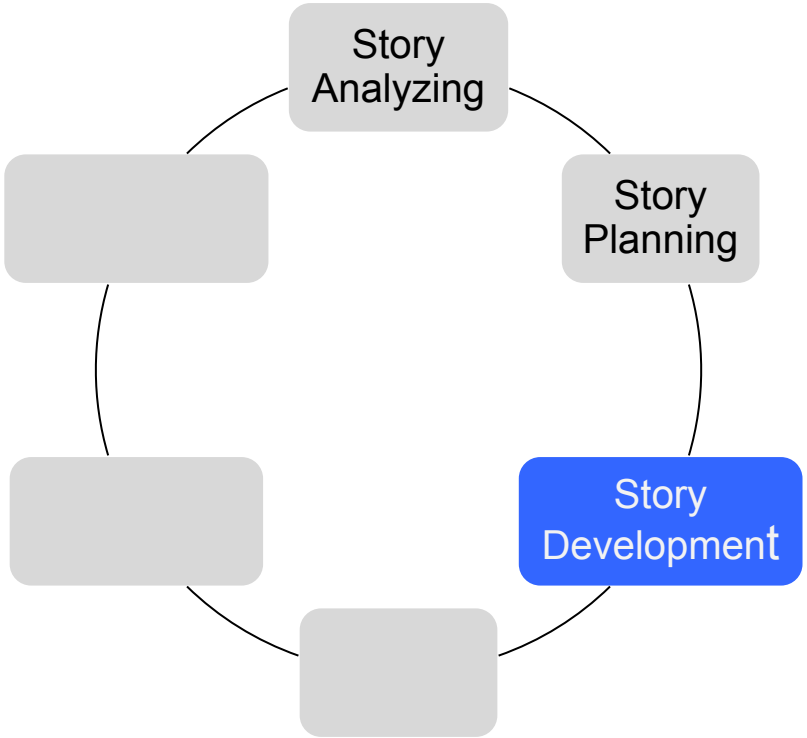




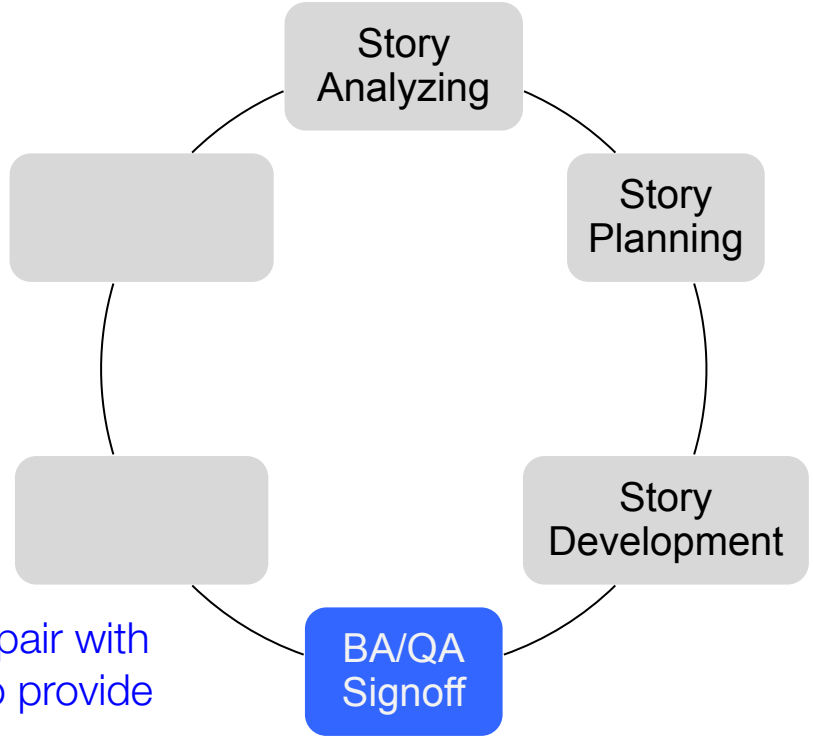
- Ask for clarification
- Business scenario and acceptance tests identification



- List down the QA tasks
- Consider the QA effort/estimates for each story estimation

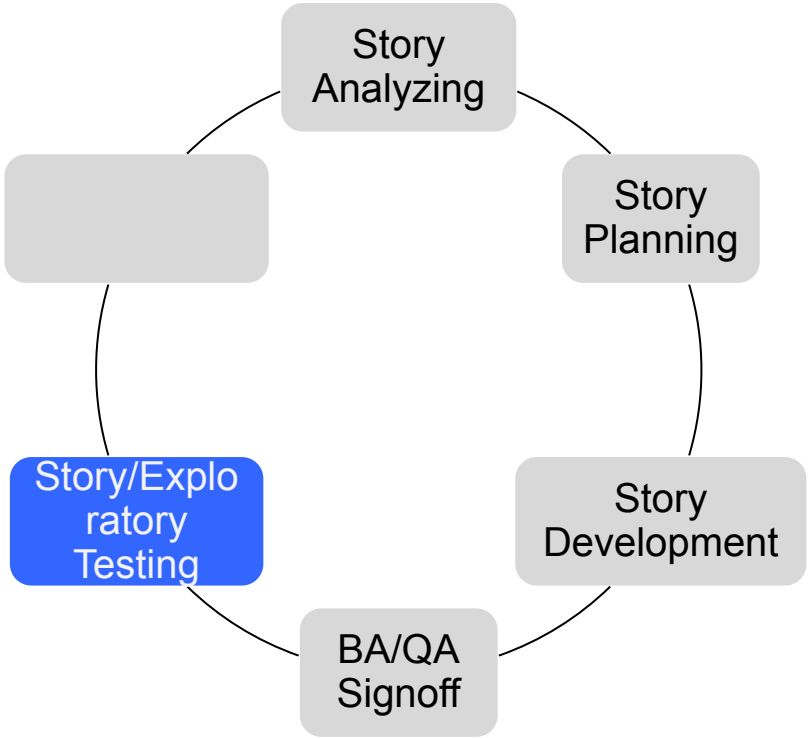


- Pair with DEV to implement automated tests
- Communicates issues/defects to the team

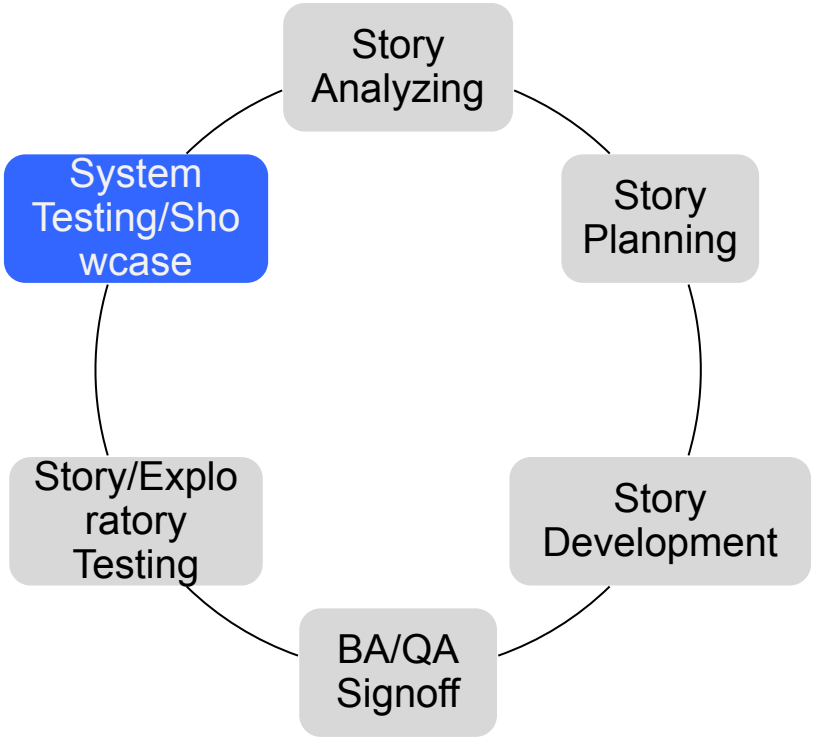


- When development complete, QA pair with BA to conduct a Dev-box testing to provide instant feedback before check-ins
- Provide feedback on unit tests/coverage

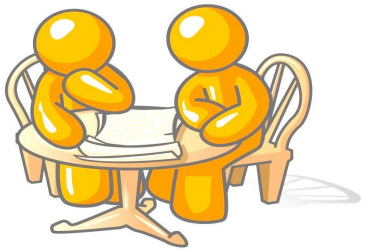
- Execute automated acceptance tests
- Exploratory testing to seek bugs in different areas/user actions
- Highlights blockers to sign-off the story
- Communicate the tests coverage to the team
- Add automated tests for bugs during exploratory testing



- Conducts end-to-end system testing
- Execute business/integration scenarios
- Highlight the team and customer, about the quality/stability of the features
- Showcase functionality/feature



Exploratory test the current stories



n

n

n+1

Pair with Dev to automate iteration n stories

Pair with BA to analyze iteration n+1 stories and write Acceptance tests

INVEST in ‘A World of Good Stories’

A good user story should be:

- "I" ndependent (of all others)
- "N" egotiable (not a specific contract for features)
- "V" aluable (or vertical)
- "E" stimable (to a good approximation)
- "S" mall (so as to fit within an iteration)
- "T" estable (in principle, even if there isn't a test for it yet)

Understanding Story

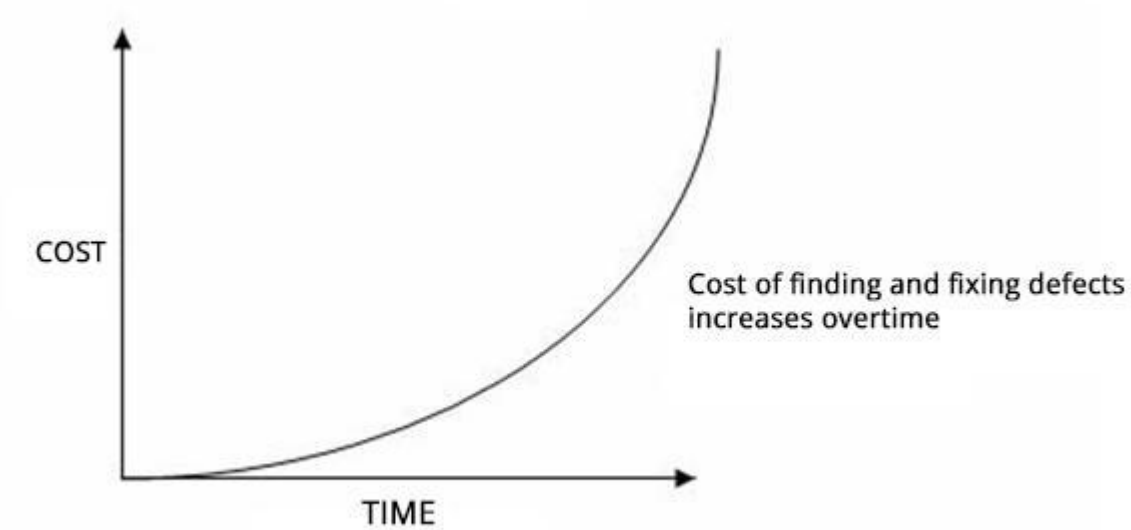
As [*role*]
I Want To [*something happens*]
So That [*goal*]

Given [*previous state*]
When [*something happens*]
Then [*new state*]

Three dimensions of QA profile

The difference between a QA and a Developer lies in the mindset.

- Business : understanding, communication skills, domain knowledge
- Technical : enforce TDD, fostering good practices for clean code, design patterns, ensuring high quality code
- DevOps : Successfully run tests on CI/CD pipeline



Picture courtesy: http://4.bp.blogspot.com/-Q3OBdVD4BKw/ULdwr_O-euI/AAAAAAAAAMU/-W7JINIicoE/s1600/sm2.gif

QUICK FILTERS: My Issues Deliverables Ideas Unestimated

To Do

GTALL-10758
Fee Evasion in Property - Editing Ad from Private to
None

Define

GTALL-10711
Pro user messaging - android checkout
Braintree on An…
None

GTALL-10718
Offline checkout journeys - android checkout
Braintree on An…
None

Ready For Dev

GTALL-10625
Provide to EMP - customer view of credits remaining
Monsters Inc.
None 3

GTALL-10281
Gumtree not showing correctly as referrer in
Motors Reporting
None 2

GTALL-10629
Reporting to EMP - Clicks on WEB URL
Monsters Inc.
None

GTALL-10668
Limit Gumtree account access for Motors Inc. users
Monsters Inc.

In Progress

GTALL-10614
Insurance category S and N motors being blocked from
None 3

GTALL-10479
Simplified Web Checkout Flow
Payment UX
None 5

GTALL-10543
Final UX tweaks - android checkout
Braintree on An…
None 3

GTALL-10710
Permissions Light-Box - android checkout
Braintree on An…
None 2

Ready for QA

GTALL-10195
Motors HPI Check - New Insurance Categories
None 3

GTALL-10323
Switch between Drop-In UI and old 'webview' payment
Braintree on An…
None 1

GTALL-10215
Submit order (production - unhappy path) -
Braintree on An…
None 2

GTALL-10474
Select Payment Type - android checkout
Braintree on An…
None 3

Done

~~GTALL-9517~~
Implement Refund functionality in Payment Service
Braintree Backend
phase7 5

~~GTALL-9518~~
Update Bapi to call payment-api for Braintree refund
Braintree Backend
phase7 8

~~GUMBUG-3803~~
Product features being removed from ads
EMP
None 3

~~GTALL-9259~~
Simplify checkout basket
Payment UX
None

Definition of Done

Swimlanes	Conditions to be met before moving to next lane
To Analyse	❏ Details of feature and requester captured in feature card ❏ BA signs up for analysis
In Analysis	❏ Business and tech analyses complete ❏ Acceptance criteria are signed off by PO+BA+QA ❏ Mock-ups of screens, supporting documents etc. (wherever necessary) are available ❏ IPM completed
Ready for dev	❏ Developers sign up for story ❏ Story kick-off completed among Devs+BA+QA
In Progress	❏ Development completed as per acceptance criteria ❏ Unit and Integration tests automated ❏ Dev-box testing done with QAs ❏ Deployed to SIT/QA env
Ready for Testing	❏ QA signs up for testing

Definition of Done

Swimlanes	Conditions to be met before moving to next lane
In Testing	❑ Tested in SIT/QA env by QAs ❑ Added automated tests ❑ Added manual test scenarios ❑ Deployed in UAT
Ready for Acceptance	❑ Product owner assigned for testing
In UAT	❑ Tested by Product owner

Quality Checkpoints

Story planning:

- ❏ There needs to be an agreement between Business and BAs to make sure the requirement is well-understood
- ❏ Descriptions, acceptance criteria, references etc. need to be in place
- ❏ Each story needs to be INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable)
- ❏ Every story in its final form should be signed off with the Business user

Story review & write scenarios:

- ❏ QAs help review the acceptance criteria and make sure all the required information for testing and test setup are covered.
- ❏ QAs also write test scenarios that can be used during development

Iteration planning:

- ❏ The team understand the stories in backlog and any missing assumptions, in-scope, out-of-scope information, mockups, references are captured on the card
- ❏ Based on estimations and team velocity, stories are slotted for iterations

Implement functionality:

- ❏ Devs can have test scenarios from QAs. They act as inputs for unit and integration tests.

Dev-box testing:

- ❏ BAs, QAs, and UX (if available) are a part of the dev-box testing. All the acceptance criteria should have been met.
- ❏ The testing needs to be as extensive as possible, so that all feedback that need to be incorporated can be done by the devs immediately
- ❏ QAs also review the coverage on unit and integration tests, to identify which tests cannot be covered at those layers (maybe for a tech constraint). These will need to be covered on upper layers of the test pyramid.

Implement acceptance criteria:

- ❏ Dev and QA pair to write the functional tests for acceptance criteria. This ensures a proper test framework, with a good design, thus making the automation maintainable
- ❏ Any additional technical support needed to get the scenarios to execute on the CI are also taken care of

Story testing:

- ❏ QAs perform an exploratory testing on an integrated environment
- ❏ QAs add test scenarios beyond the acceptance criteria in automation or manual repositories, as required
- ❏ If there are any acceptance criteria which is not met, the story is moved back to development
- ❏ If regressive defects were found, they are raised as defects

Business showcase:

- ❏ The Product owner and all stakeholders need to agree with what is built.
- ❏ The frequency of the showcase can be decided by the team. But the more often we have it, the chances of digressing from the requirements are lesser

UAT:

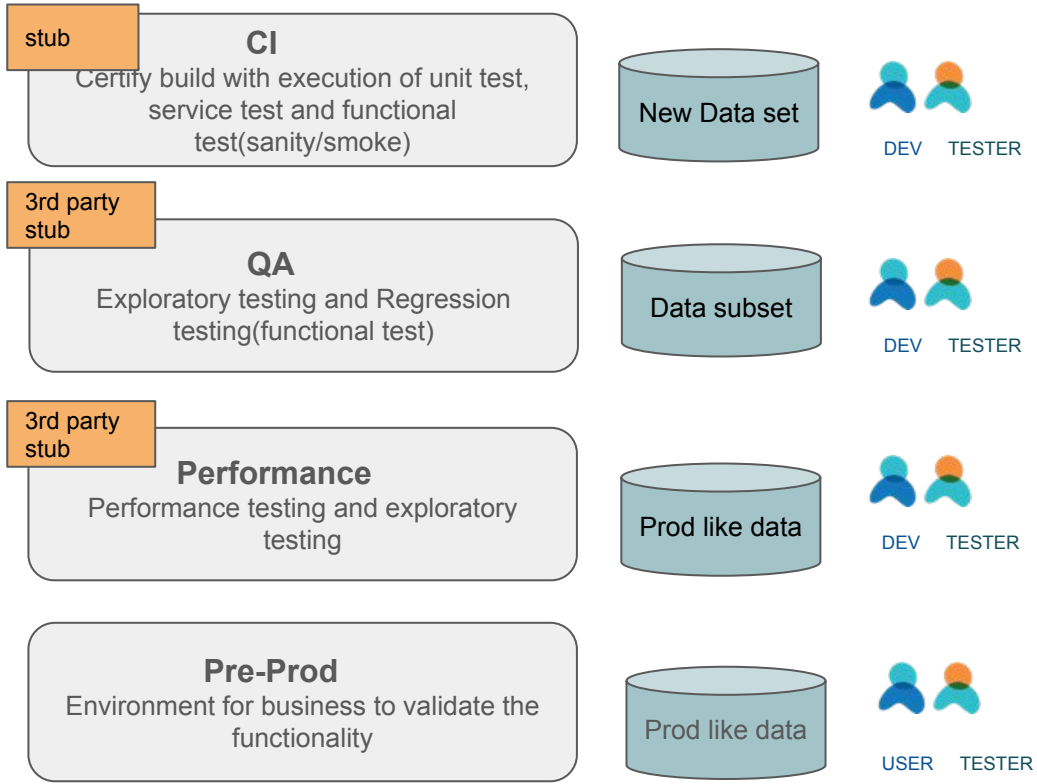
- ❏ Exploratory testing by Beta clients ensures we are in-line with usability requirements.
- ❏ Additional requirements can be captured as new stories/enhancements

Environments & Deployments

Different environment serves different purpose

Lack of environments could affect:

- Task, effort of a team
- feedback
- minimal or no testing on different volume of data
- Visibility on the results achieved



Types of Testing

TYPES OF TESTING	AUTOMATED / MANUAL	ENVIRONMENT	EXECUTED BY	COMMENTS
During Development				
• Unit, component, integration	Automated	Development	CI	Continuous integration, faster feedback
• Deployment	Automated	Development	CI	Continuous deployment
Functional testing				
• Smoke	Automated	CI / QA	CI	Is the app up?
• Sanity	Automated	CI / QA	CI	Sanitise major functionalities
• Regression	Both	QA/Regression env	Manual – QA, Regression team Automation - CI	
• Exploratory	Manual	QA env	QA	
Compatibility testing				
• Browser	Both	QA/Regression env	Manual – QA Automation - CI	
• OS	Both	QA/Regression env	Manual – QA Automation - CI	
Usability testing	Manual	UAT env	QA/BA + Business users	

Types of Testing (continued)

TYPES OF TESTING	AUTOMATED / MANUAL	ENVIRONMENT	EXECUTED BY	COMMENTS
Performance testing				
• Response time performance	Both	Dev environment (benchmarking), QA env (comparative analysis), Pre-prod/UAT env		Needs prod-like infra setup for generating reliable results
• Load testing	Both	Pre-prod/UAT env		Needs prod-like infra setup The environment should be isolated
Security testing				
• Development security testing	Both	Dev/QA env	Manual – QA+Dev Automation - CI	Incorporated in daily development including story acceptance criteria
• Pen-testing	Both	Pre-prod/UAT env	Pen-testers	
User Acceptance testing	Manual	UAT env	Business users	

BUG BASH

DEPLOY & TEST

Observations

Observation	Recommendation
Late feedback on functionality developed.	• Dev box Testing • Run end-to-end smoke test on CI • Run functional tests on Dev box
Lack of visibility on the quality of the application to stakeholders	• Broadcast summary level view of defects and test report regularly • Go/No-Go meetings before release
Collaboration within a team	• Dev/QA write and maintain automated tests • QA/Dev collaborate in writing acceptance criteria • Include QA in planning, User Story definition, and dev kick-off
Inadequate Safety Net across all levels of the application (Functional and Cross Functional)	• Review definition of “Done” • Build safety net for each service before redesigning it • Build end-to-end functional tests • Fix Brittle/Ignored functional tests
Testability not built in to the system *Example: Payments, and IVR flows, can only be tested in Production*	• Stubs for third party integrations • Testability built in to story development

In Summary...

- Quality on it’s own isn’t meaningful. Meeting the needs of users and business is meaningful. QAs advocate for processes that ensure a product meets these needs of the users and the business

- Aim to push conversations about risk upstream by continuously shortening feedback loops

- There are lots of hats a QA can wear - you can forge your own path of which you want to dive into next

Find out more

- Becoming a QA Leftie, by Kenny Cruden. Move the conversations upstream
- So what is a QA?, by Sarnacke, Kevin, Abby and Tim. QA Hats!
- Pride and Paradev, by Alister Scott. A book of agile testing contradictions