

Agile Automation

What does “Agile” automation mean?

Rob Manger
Senior Quality Analyst



REAGroup

Who am I?

- Background in Quality Assurance
- Software development using Agile and Lean
- Passionate about process and efficiency improvement
- Worked at:



What am I talking about today?



Why do we test?

- Make sure the product meets expectations
- To ensure that the system is robust
- To mitigate risk of failure
- Gain confidence in the product



Why do we automate?

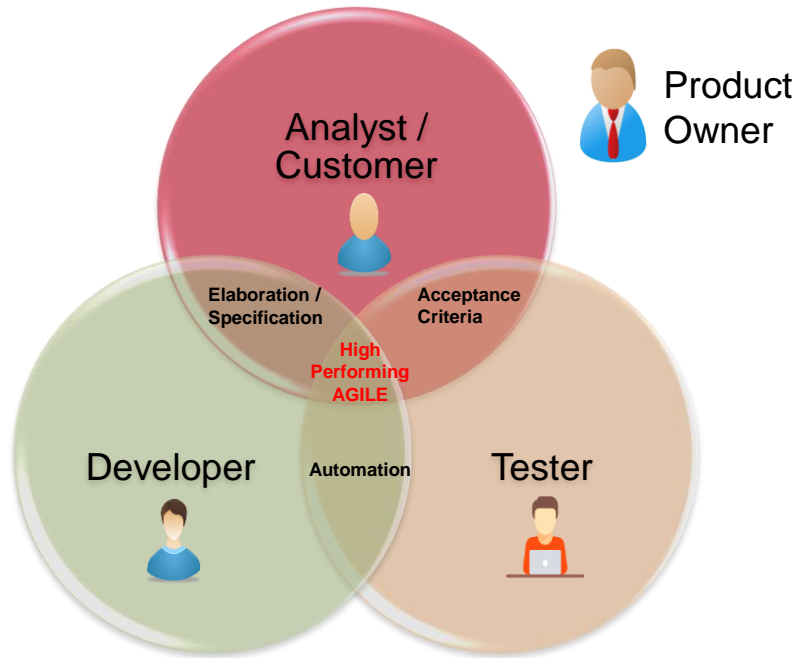
- Remove manual repetition
- Speed up the feedback loop
- Enable confidence in the product before releasing
- Enable manual testing to focus on where it is needed



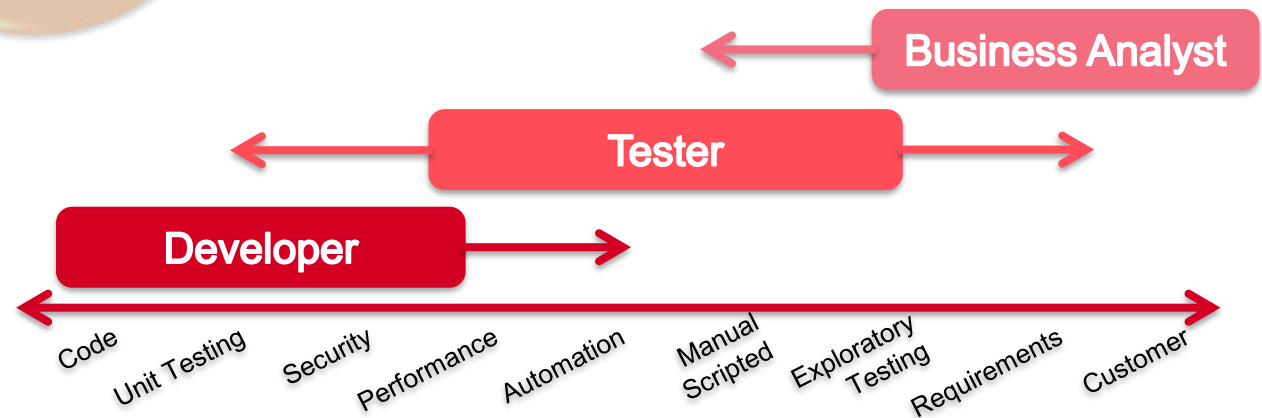
What effect does Agile have?

- What flavour of Agile? Lean? XP? RAD?
- Shorter lifecycle between requirements definition and delivery
- Nature of Agile is 'adaptive to change'
- Less time for up front framework design and creation
- Less predictability in an evolving environment
- Increased dependency on *rapid* regression testing
- Reduced dependency on monolithic frameworks
- Team's group ownership of Quality

Who does the automation?



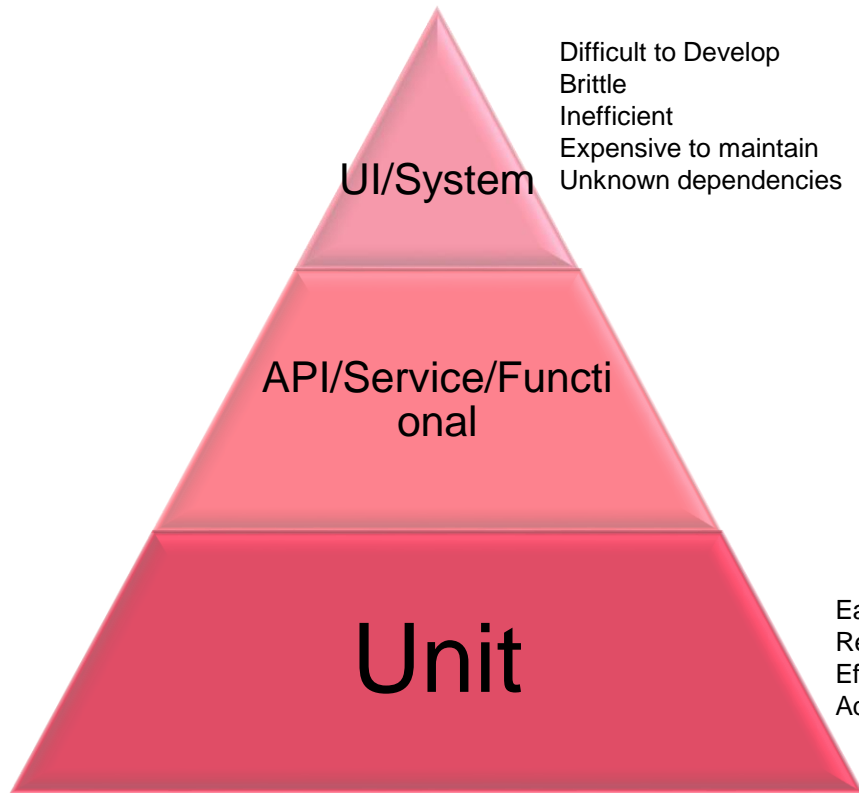
- QA vs Dev vs Dedicated automation team?
- Build Quality In – no separation of dev and test effort
- The story isn't 'Done' until it is tested accepted
- QA = Quality Assistance
- “Developers own the Acceptance Tests” – Dave Farley



What should we automate?



What should we automate?

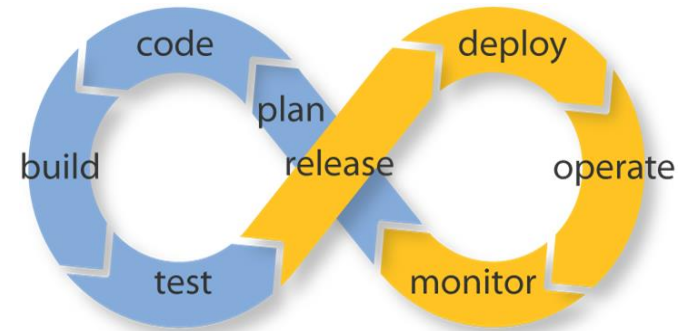


- How much is enough?
- Is 100% test automation viable?
- Value Based approach
 - Critical Path – Highest Value
- Test at the right level
- Focus on “What” not “How”
- Keep your tests small
 - Quicker to write and run
 - Easier to maintain
 - Easier to identify what the problem is
- Review tests at all levels
 - Make sure they are testing the right thing

When do we automate?

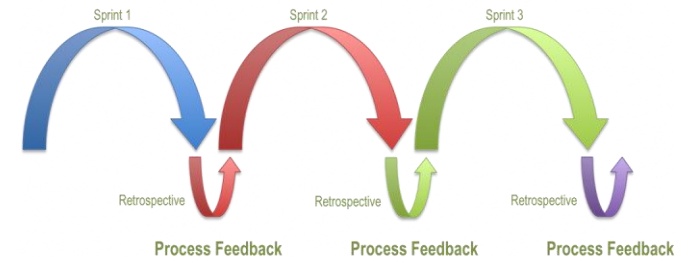
Planning and Writing

- As early in the development process as possible
- TDD has automation written before development
- BDD has automation occurring at the same time as requirements definition
- Don't separate test effort from dev effort



Execution

- Execute Locally, before every check-in/commit
- Continuous Integration / Deployment means every build
- Parallel execution enables running more tests quicker



How do we automate?

Pre-requisite: Make sure the SUT is testable

- Make sure the toolset chosen is Fit for Purpose
- BDD is perfect for some scenarios, not for all
- TDD is perfect for technical people, but not for everyone
- Contract driven development – automated integration testing at a unit test level
- Use stubbing as much as appropriate

Too Much?



Too Little?

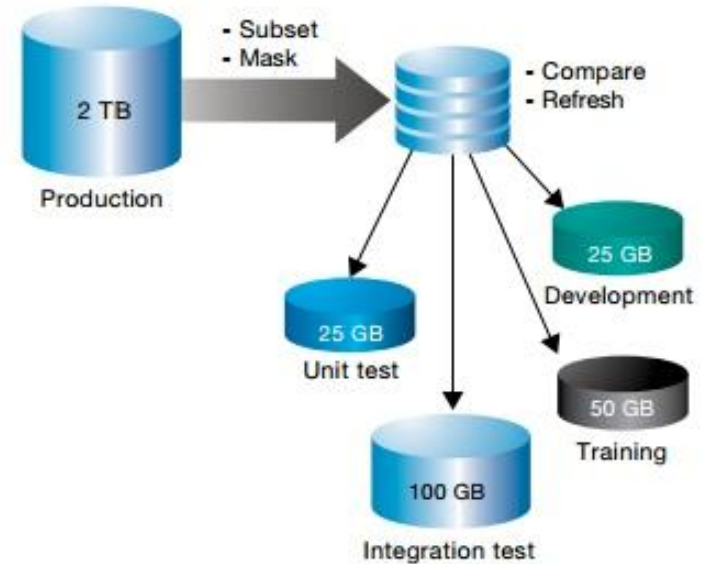


```
Feature: 3355 As A Jobseeker I want a simplified registration with email and password
  As a Jobseeker
  I want a simplified registration with email and password
  So that I can register more easily
```

```
Scenario: 01 - Jobseeker views registration (In process)
  Given the Jobseeker is logged out
  And the in process registration has been initiated
  When registration is displayed
  Then registration contains an Email field
  And a password field
  And remember me is checked by default
  And forgotten login details link is displayed
```

How do we automate? (cont..)

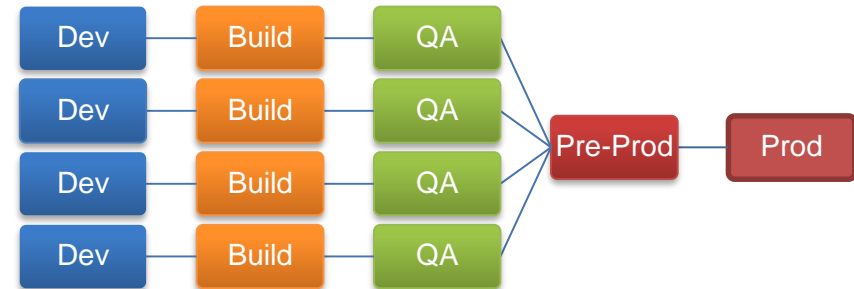
- Create Atomic tests – self reliant, can run anywhere, are repeatable
- Test Data management
 - To create or to prepare
- Setup, verification and tear-down through the UI layer is expensive. Other options? e.g.:
 - Direct API access
 - Code/DLL hooks
 - Headless execution
 - Direct DB access (if no other option)



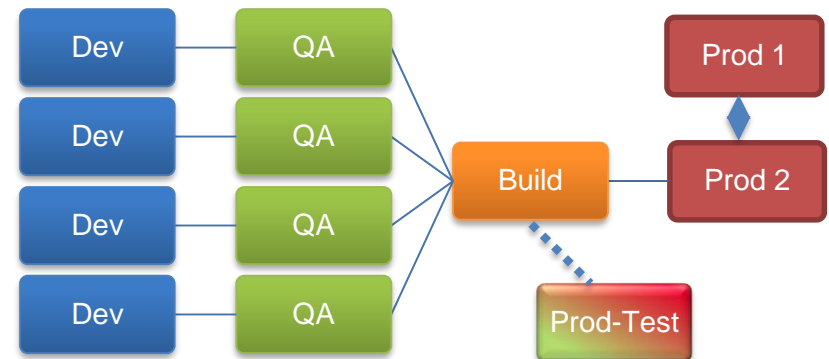
Where should we automate?

- “Production-like” environment
- Decision affected by:
 - Solution Architecture – i.e.: Micro-services vs. Monolithic
 - Difference between Dev and Prod environments
 - Number of different integration points
 - Data Access restrictions
 - Cloud vs Physically hosted
- Environment Management (eg: Who owns a shared staging environment?)
- QA pairing with DevOps – confidence in deployment process
- “Cattle, not Pets”

Shared Environment – Manual Deployment



Up Front QA – Continuous Deployment



Closing thoughts

- Automation should be BAU
- Make sure it's Fit For Purpose
- Don't lose site of the reason you are automating
- Make sure your automation is focused at the appropriate level
- Make sure the results are visible
- Automation of build/deployment process equally important
- Alerts and Monitoring are also equally important

- **Have fun doing it!!**

Questions



Robert Manger
rob.manger@rea-group.com
@robertmanger