Building Quality In...

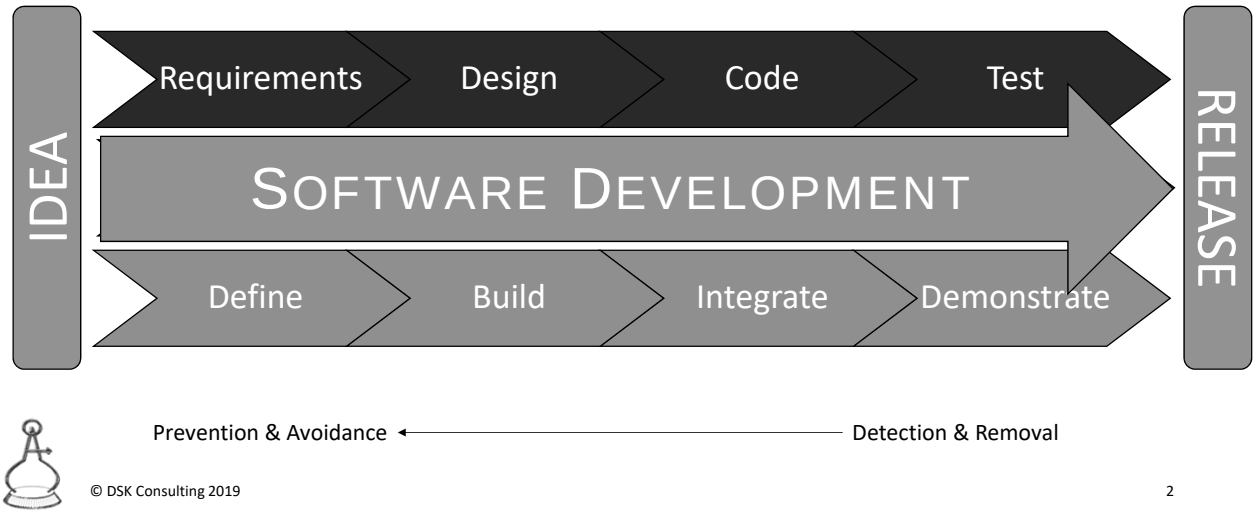
..By Engaging To The Left

Darin Kalashian
SSGB, CSM, CBRP, CQMgr, CSQE

SQGNE Software Quality Group of New England

September, 2019

What Does "Quality To The Left" Mean?



2

Expectations of Quality Engineering?

What do our executives want from our QE individuals?

Where do leaders think QE can add value?

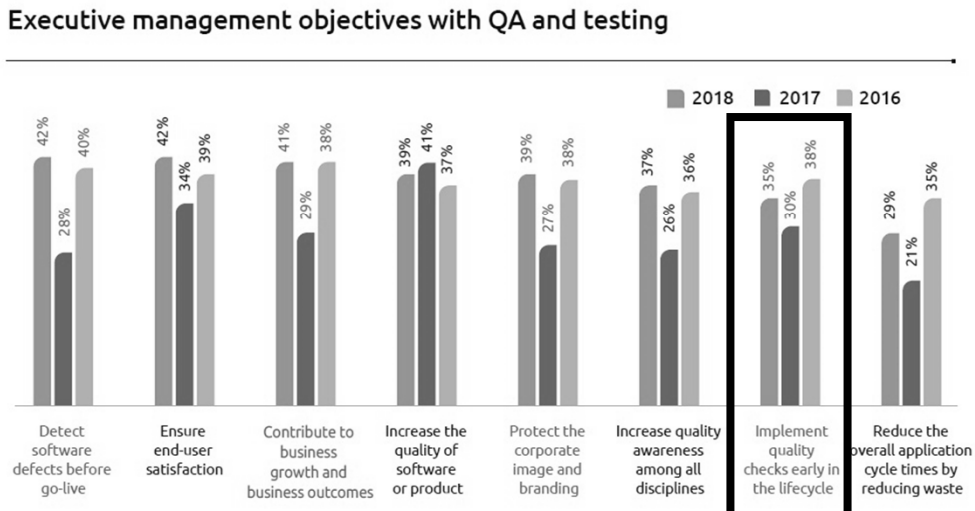
What will our leaders fund and help drive?

 3

Expe

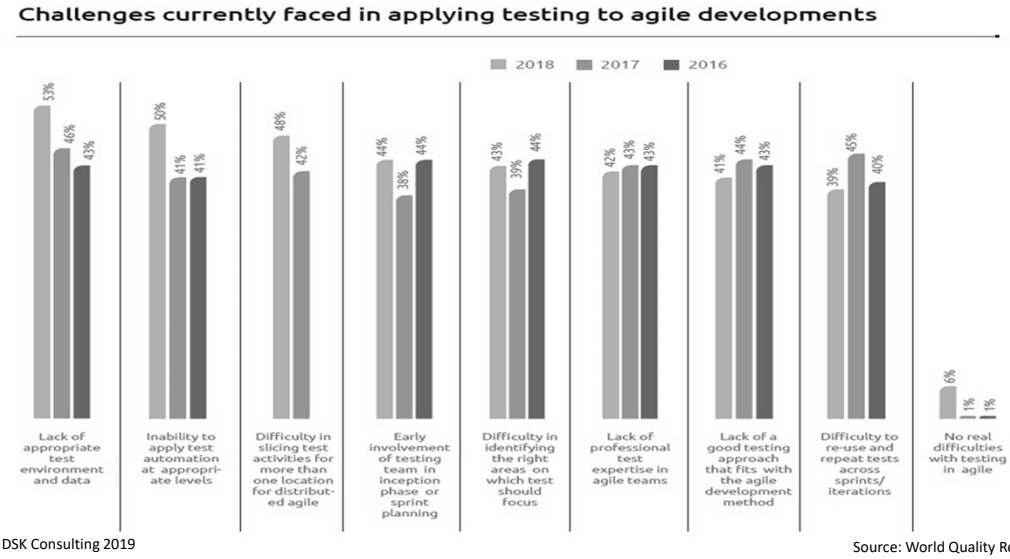
 4

Where is the Value?



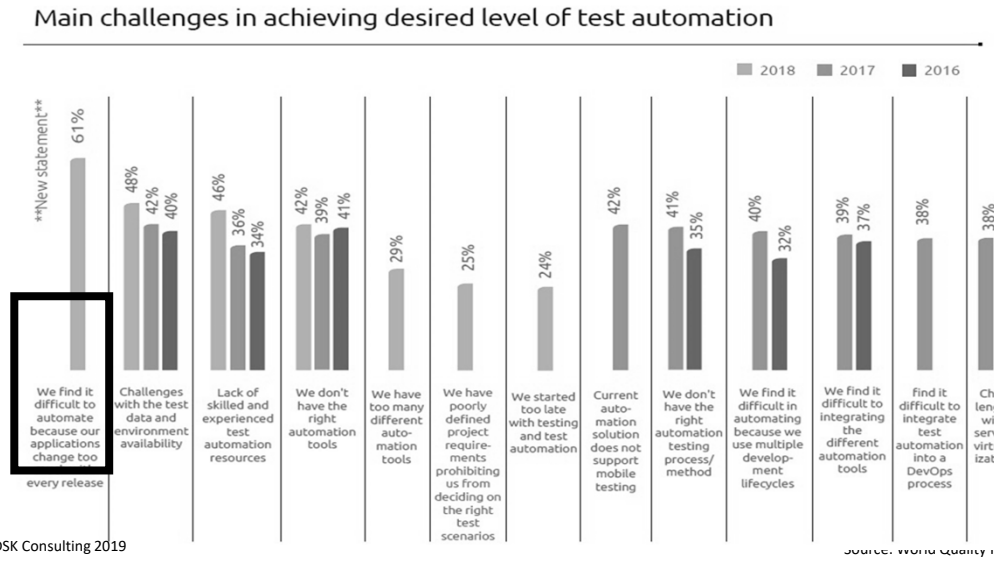
Source: World Quality Report: Vol. 10 5

What Are The Challenges?



Source: World Quality Report: Vol. 10 6

How Do We Improve Impact?



Source: World Quality Report: Vol. 10

Quality Take-aways

"Organizations are unable to keep pace with the volume and frequency of testing required" – Mark Buenen, VP, Sogeti

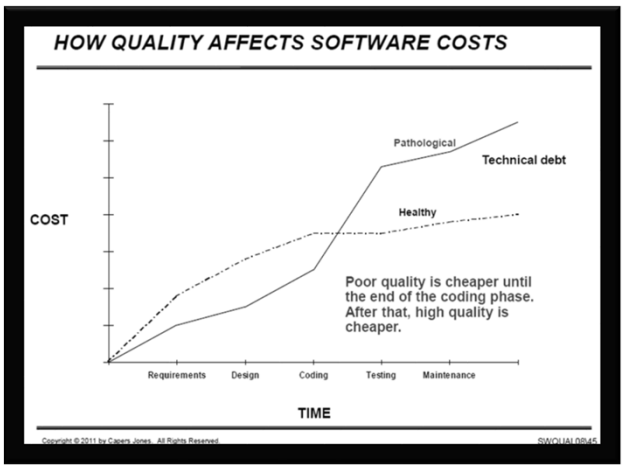
"Many organizations are retaining a thin layer of centralization known as the community of QA practice or Test Excellence Center as it helps them strike an optimal balance between flexibility, self-running teams and the ability to industrialize best practices and resources"

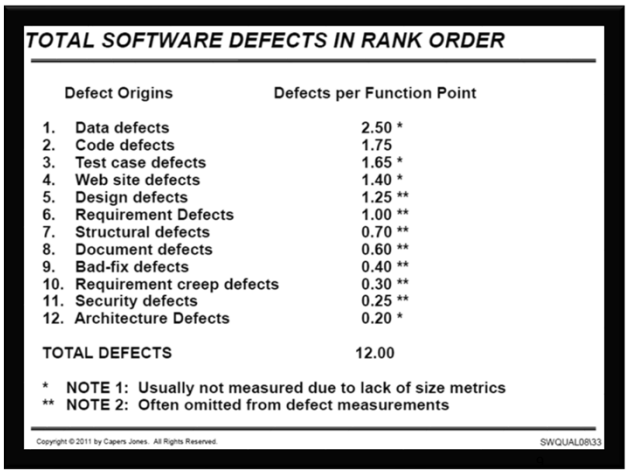
– Ajay Walgude, VP, Capgemini

"The role of quality assurance and testing has changed from mere defect finding to one of being an enabler of customer satisfaction and business outcomes."

– Mark Buenen, VP, Sogeti

Impacting Software Quality





TOTAL SOFTWARE DEFECTS IN RANK ORDER

Defect Origins	Defects per Function Point
1. Data defects	2.50 *
2. Code defects	1.75
3. Test case defects	1.65 *
4. Web site defects	1.40 *
5. Design defects	1.25 **
6. Requirement Defects	1.00 **
7. Structural defects	0.70 **
8. Document defects	0.60 **
9. Bad-fix defects	0.40 **
10. Requirement creep defects	0.30 **
11. Security defects	0.25 **
12. Architecture Defects	0.20 *
TOTAL DEFECTS	12.00

* NOTE 1: Usually not measured due to lack of size metrics
** NOTE 2: Often omitted from defect measurements

Copyright © 2011 by Capers Jones. All Rights Reserved. SWQUAL08#33

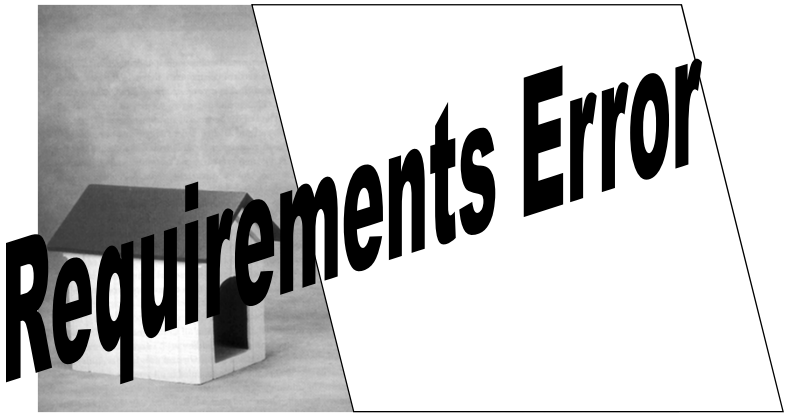
Tonight's Objectives

1. ***Investigate what activities and tools can be utilized "to the left" to build quality in and prevent issues "to the right"***
   - building a requirements/story quality dashboard
   - using requirements and design modeling to assess completeness
2. ***How to strengthen your testing strategy***
   - Using TDD while defining a multi-faceted test strategy
3. ***Get QE out of the policing role and more into an enabler role.***
   - using continuous integration to expedite time to market

Testing Quality Into Requirements/Stories

 11

Testing Requirements/Story Quality

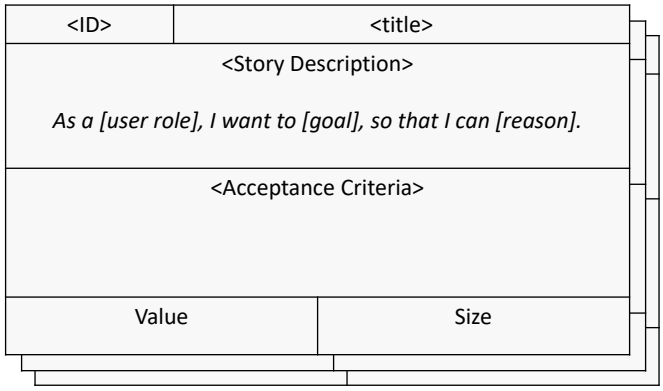


©1998 Technology Builder Inc.

 12

Building On A Good Foundation

- The following is suggested as a best-practice for defining a Story (a.k.a. Requirement)



STARTING PLACE:

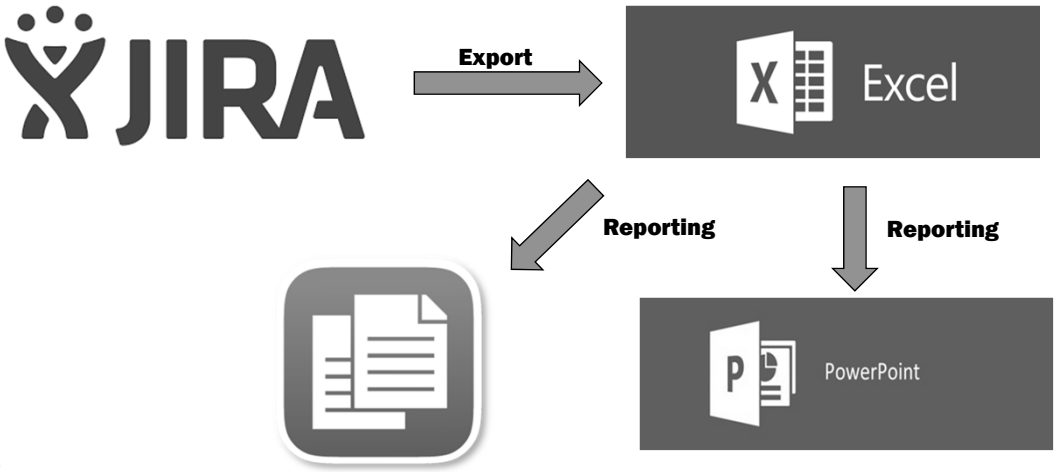
- Story Description
- Acceptance Criteria
- Size
- Ranking Order

Story Quality Measures

Define criteria to give some indicator of User Story quality:

Quality Attribute	Desired Behavior	Source	Measure
User Story Frame	Have story framed such that the **role**, desired **operation**, and the intended **benefit** are clearly captured.	Description: Text includes story "As <**role**>, I want to <**operation**>, so that <**benefit**>".	**Description**: All 3 : Green 1~2 : Yellow None : **Red**
User Story Size	Smaller stories are better understood and have a higher chance of being delivered on-time, on-feature, and on-quality.	Story Points	Story Points: <= 8 : Green 9~14 : > 15 : **Red**
User Story Boundaries	Acceptance criteria defines the minimum viable use case which the story must deliver on.	Acceptance Criteria:	**Acceptance Criteria**: [defined / **not defined**]
User Story Rank	Have stories ranked from most important to least important for the order in which the team should work the list.	Ranking	Order is logical and defines Most-important/High-value to Least/Low-value

Getting To A Dashboard



 15

Story Quality Dashboard (Before)

					STORY QUALITY		
Issue key	Sprint	Size	Summary	Scrum Team	Framed	Sized	Bound
SUPRA-43950	Sprint 7.1	13	Engine Call Home with heartbeat via SWN	Andretti			
SUPRA-41881	Sprint 7.1	23	In vehicle CPR updated via bluetooth channel	Andretti			
SUPRA-41884	Sprint 7.1	5	Encrypt server data with AES256 keys	Andretti			
SUPRA-42134	Sprint 7.1	10	Implement telemetry data reporting	Andretti			
SUPRA-39241	Sprint 7.2	5	Update CAN bus to 12.1 specification	Andretti			
SUPRA-41882	Sprint 7.2	8	Heads up display of diagnostic data	Andretti			
SUPRA-41883	Sprint 7.2	15	Voice enabled diagnostics	Andretti			
SUPRA-41943	Sprint 7.2	1	Delete driver profile stored on Support Server	Andretti			
SUPRA-42919	Sprint 7.2	0	Braking controller coef modification	Andretti			
SUPRA-42870	Sprint 7.2	13	[SPIKE] Investigate telemetry data to collect	Andretti			
SUPRA-42452	Sprint 7.2	3	Improve steering feedback response	Andretti			
SUPRA-42133	Sprint 7.2	5	Create driver profile in Suport Server	Andretti			
SUPRA-31089	Sprint 7.2	3	Optimize emissions to be 98% clean burn	Andretti			
SUPRA-19218	Sprint 7.2	15	Safety parachute automated deployment	Andretti			
SUPRA-42020	Sprint 7.3	3	Read/display driver profile on Support Server	Andretti			
SUPRA-41944	Sprint 7.3	8	Udpate driver profile stored on Support Server	Andretti			
SUPRA-43620	Sprint 7.3	15	[SPIKE] Research fuel pump configuration to suppc	Andretti			
SUPRA-43550	Sprint 7.3	12	In car indicators for vital monitors	Andretti			
SUPRA-41912	Sprint 7.3	23	Backup support server data to cloud storage	Andretti			
SUPRA-44034	Sprint 7.3	0	Increase Engine Max RPM	Andretti			

 16

Story Quality Dashboard (After)

					STORY QUALITY		
Issue key	Sprint	Size	Summary	Scrum Team	Framed	Sized	Bound
SUPRA-41881	Sprint 7.1	5	In vehicle CPR updated via bluetooth channel	Andretti	●	●	●
SUPRA-41884	Sprint 7.1	5	Encrypt server data with AES256 keys	Andretti	●	●	●
SUPRA-42134	Sprint 7.1	10	Implement telemetry data reporting	Andretti	●	○	●
SUPRA-69418	Sprint 7.1	5	[SPIKE] Investigate Call Home Technology	Andretti	●	●	●
SUPRA-39241	Sprint 7.2	5	Update CAN bus to 12.1 specification	Andretti	●	●	●
SUPRA-41882	Sprint 7.2	8	Heads up display of diagnostic data	Andretti	●	●	●
SUPRA-41883	Sprint 7.2	14	Voice enabled diagnostics	Andretti	●	○	●
SUPRA-41943	Sprint 7.2	1	Delete driver profile stored on Support Server	Andretti	●	●	●
SUPRA-42919	Sprint 7.2	0	Braking controller coef modification	Andretti	●	●	●
SUPRA-42870	Sprint 7.2	8	[SPIKE] Investigate telemetry data to collect	Andretti	●	●	●
SUPRA-42452	Sprint 7.2	3	Improve steering feedback response	Andretti	●	●	●
SUPRA-42133	Sprint 7.2	5	Create driver profile in Suport Server	Andretti	●	●	●
SUPRA-31089	Sprint 7.2	3	Optimize emissions to be 98% clean burn	Andretti	●	●	●
SUPRA-19218	Sprint 7.2	0	Safety parachute automated deployment	Andretti	●	●	●
SUPRA-42020	Sprint 7.3	3	Read/display driver profile on Support Server	Andretti	●	●	●
SUPRA-43950	Sprint 7.3	3	Engine Call Home with heartbeat via SWN	Andretti	●	●	●
SUPRA-41944	Sprint 7.3	8	Udpate driver profile stored on Support Server	Andretti	●	●	●
SUPRA-43620	Sprint 7.3	0	[SPIKE] Research fuel pump configuration to suppo	Andretti	●	●	●
SUPRA-43550	Sprint 7.3	12	In car indicators for vital monitors	Andretti	●	○	●
SUPRA-41912	Sprint 7.3	23	Backup support server data to cloud storage	Andretti	●	●	●
SUPRA-44034	Sprint 7.3	0	Increase Engine Max RPM	Andretti	●	●	●

Benefit:

1. Work that is well understood is positioned first.
2. Easy to understand were the risk is – practice risk mitigation techniques.
3. Spikes, while not being framed, are sized and bound so that we understand what needs to be accomplished as part of the effort – not open ended discovery.

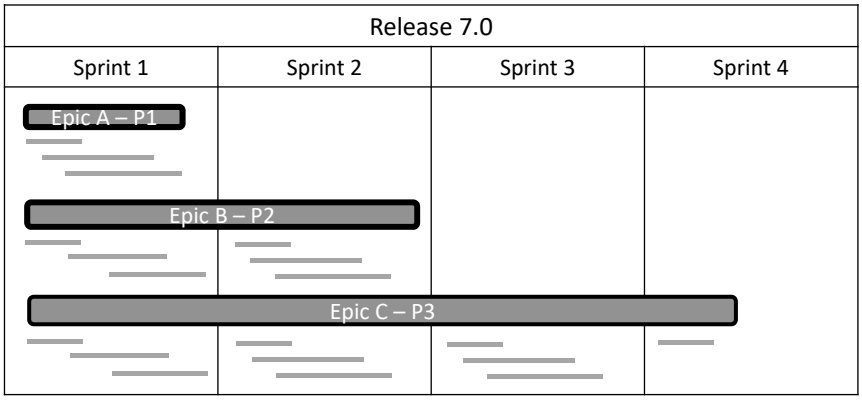
17

Roadmap To Quality

- Understand if the sequencing of stories/requirements gets you to something that is releasable and useful and has value.
- Use the “Container” model to associate stories/requirements that need to be delivered together (Epics, Stories)
- Constrain User Stories to Sprints, Epics to Releases
- Limit Too many epics/stories in progress (WIP)
- Build a Sprint/Release map to see the release

18

Testing the Release Map



EPIC T-SHIRT SIZING:

Small: <50% (1~2 Sprints to deliver)
Medium: 50% ~ 100% (3~4 Sprints to deliver)
Large: >100% of Release Cycle
X-Large: Unknown/Unplannable

LEGEND:

Epic:
Story:

 19

Modeling the Requirements and Design

In our requirements (user stories) focused software development life cycle (SDLC), it is all to easy to lose sight of the bigger, system picture.

Example:

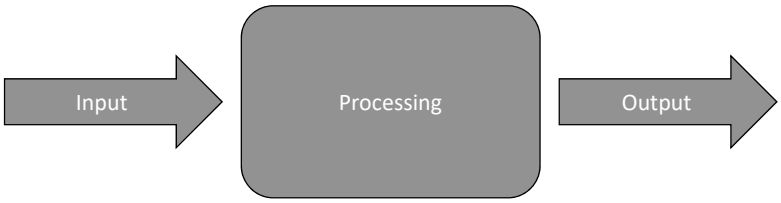
STORY 1: **TEAM A**: As the braking controller, when I detect the slip_sensor_enabled flag has been set, I want to display the ABS Active alarm to the driver, so that the driver knows ABS is engaged.

STORY 2: **TEAM C**: As the slip sensor, when I detect the wheel is slipping on the road, I want to set the slip_sensor_date with the current time, so the braking controller will know that ABS is active.

 20

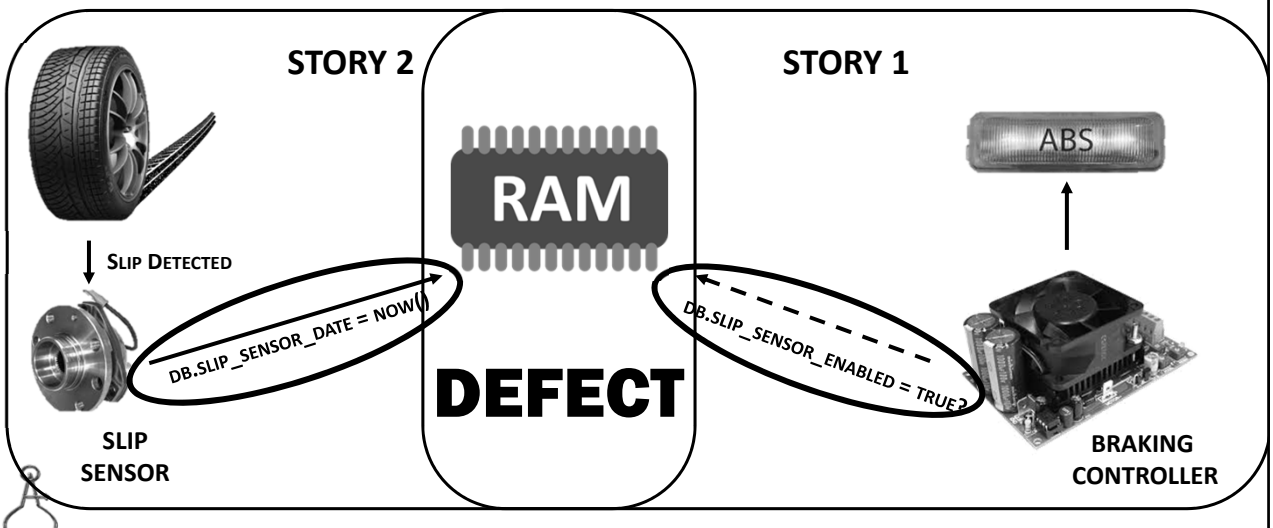
Requirements and Design Modeling

Process Specification



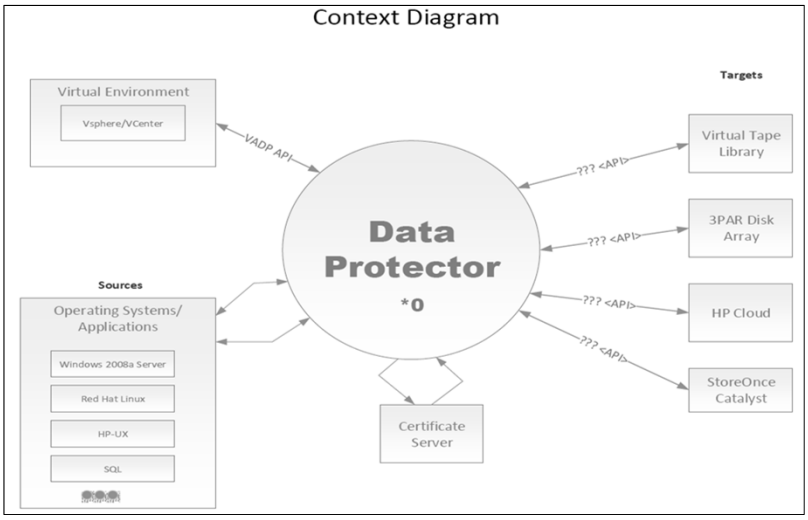
 21

Requirements and Design Modeling



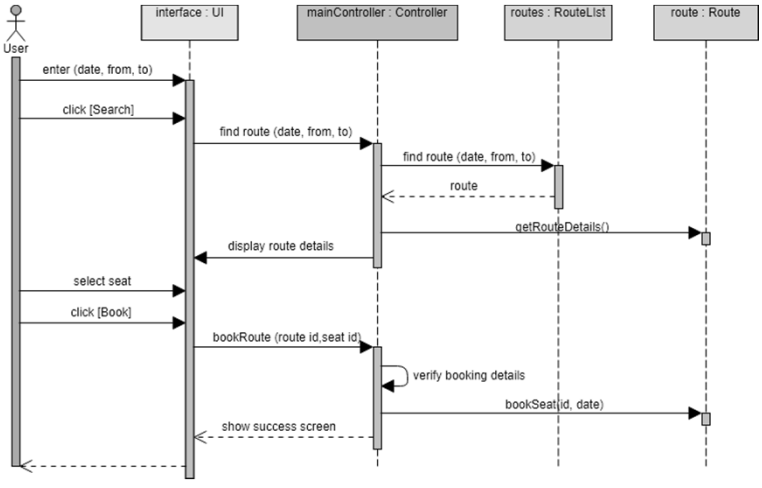
 22

Modeling - Data Flow Diagram



 23

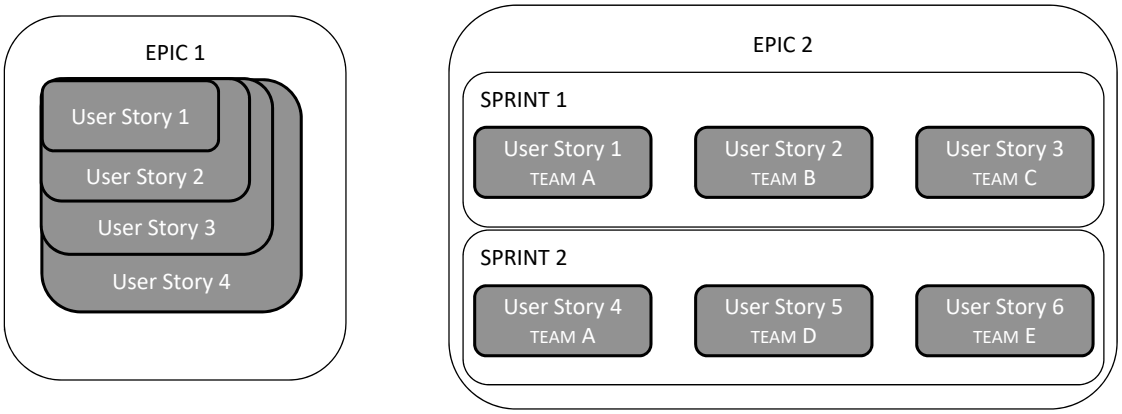
Modeling - Sequence Diagram



source: https://online.visual-paradigm.com

 24

Modeling in Agile



25

Strengthening Your Testing Impact

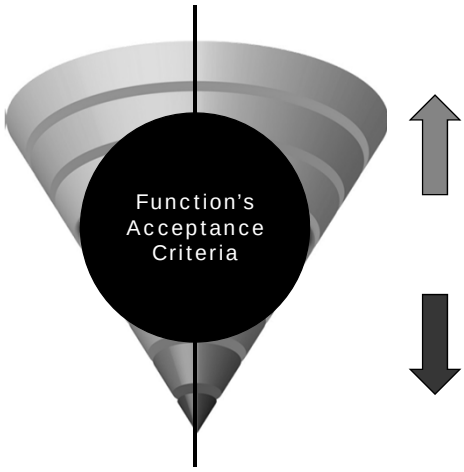
26

Building a Test Strategy

- What is the goal of your testing strategy?
  - Find bugs?
  - Verify that something works?
  - Instill confidence?

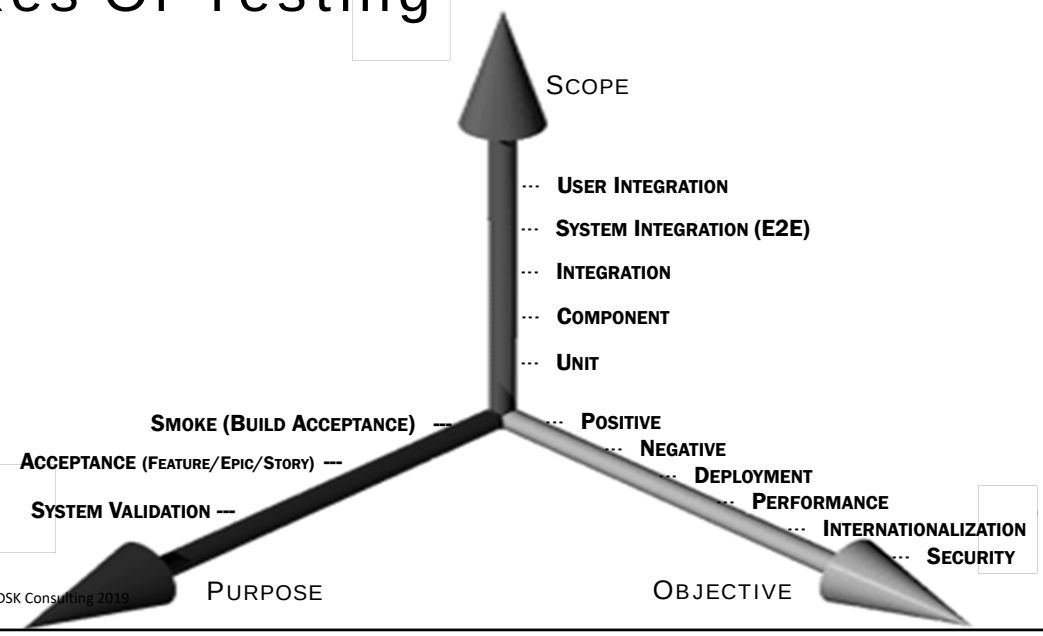
Validate that a respective function can complete it's objective

- The more it can perform unexpected functions, the higher the quality
- The less if can perform the expected function, the lower the quality



 27

Axes Of Testing



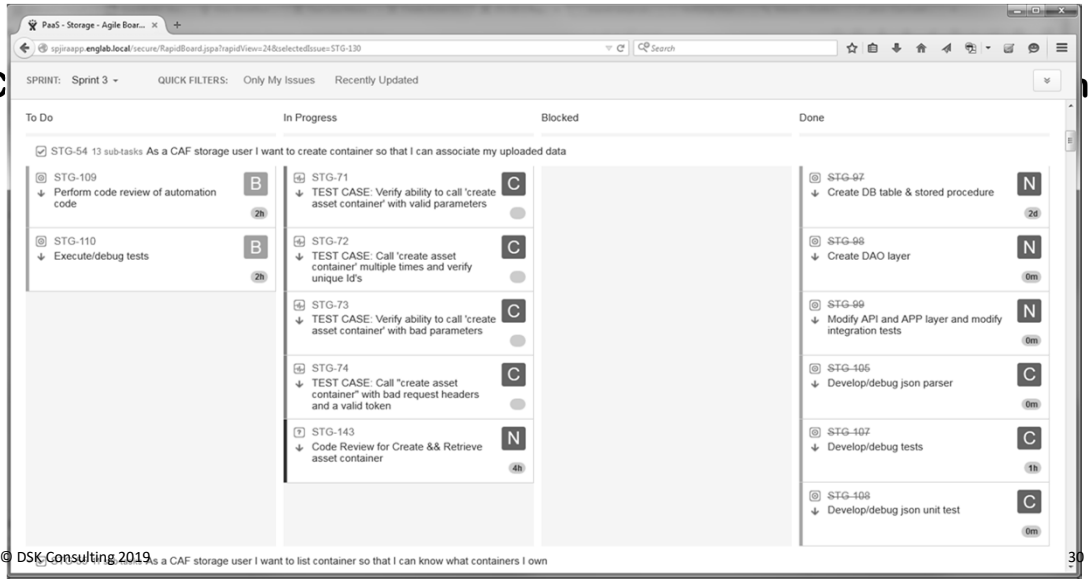
 28

Defining a Multifaceted Test Strategy

Test Case	Objective	Behavior	Scope	Purpose
TC1: ABS indicator displays	functional	Positive	unit	Acceptance
TC2: ABS indicator clears	functional	Positive	integration	Acceptance
TC3: ABS indicator luminosity	performance	Positive	User integration	System Validation
TC4: ABS in extreme heat (in spec)	stress	Positive	system	System Validation
TC5: ABS in extreme cold (out of spec)	stress	negative	system	System Validation

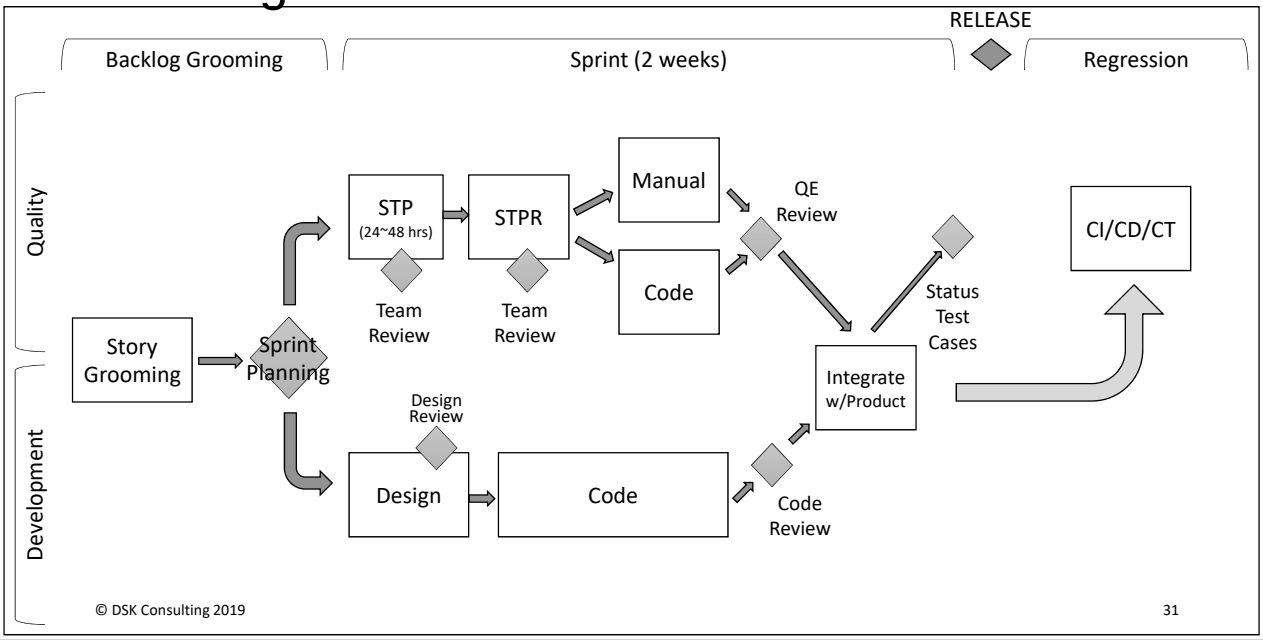
29

Integrating Tests In

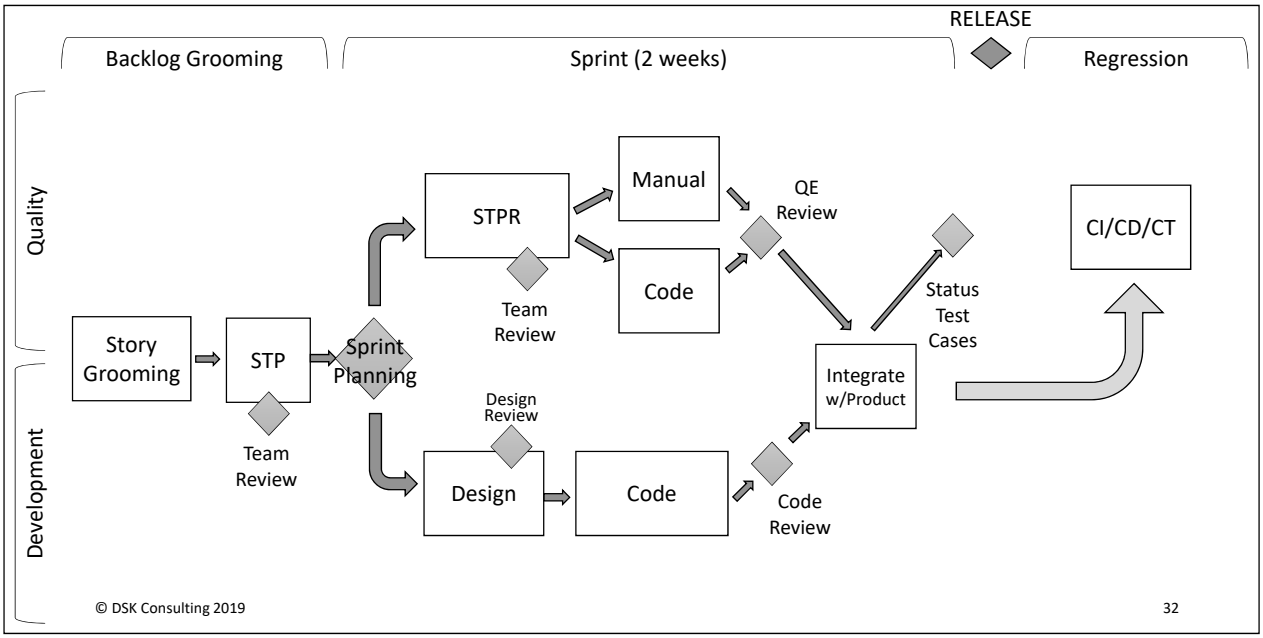


30

Defining the Test Workflow



A Better Test Workflow



Continuous Integration / Continuous Deployment / Continuous Test Model

Quality Becoming an Enabler

Do you ever feel as if this is your role?

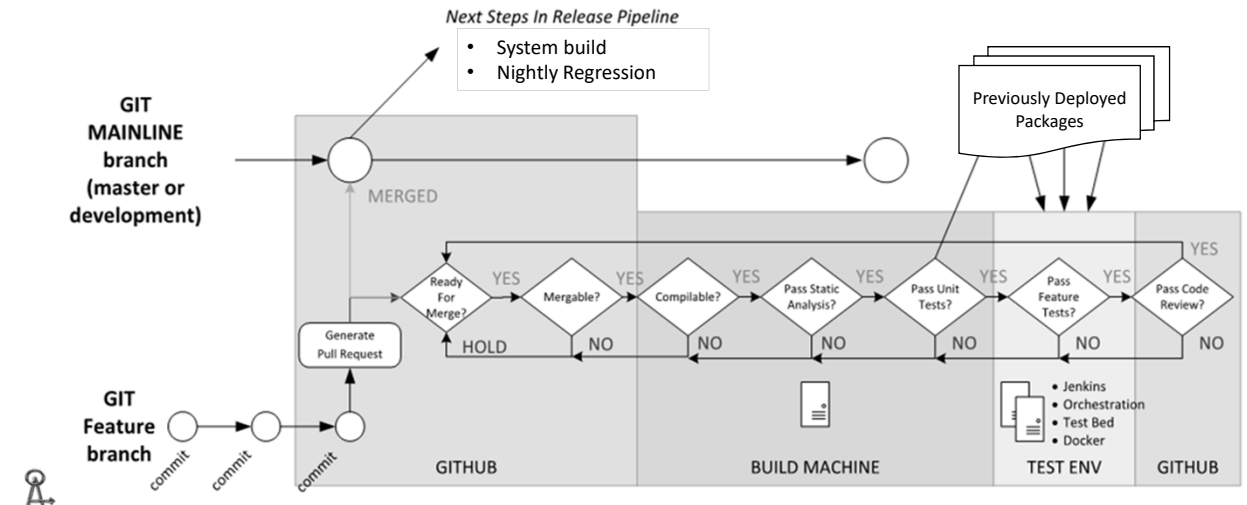
With the Continuous Integration, Deployment, and Test model, Quality Engineering can now let the infrastructure police product development.

Quality Engineering is now a contributor the infrastructure (like the other roles)

- Tests
- Data
- Environments/Configuration

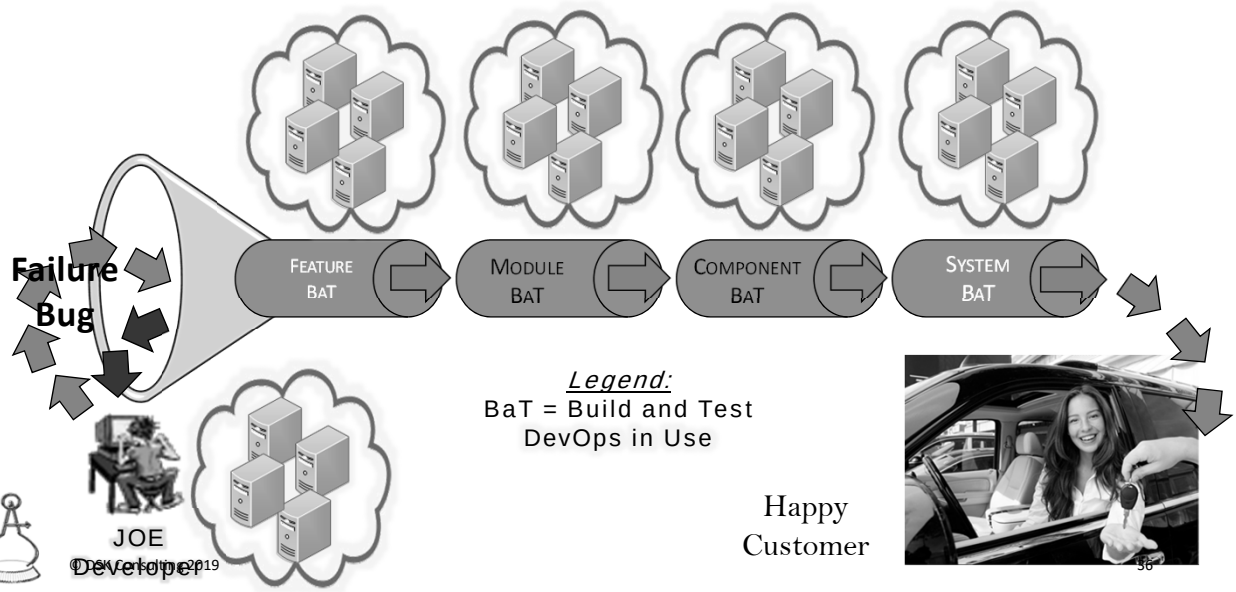
source image: VectorStock

Building a Continuous Integration Model



 35

A CI/CD/CT Pipeline

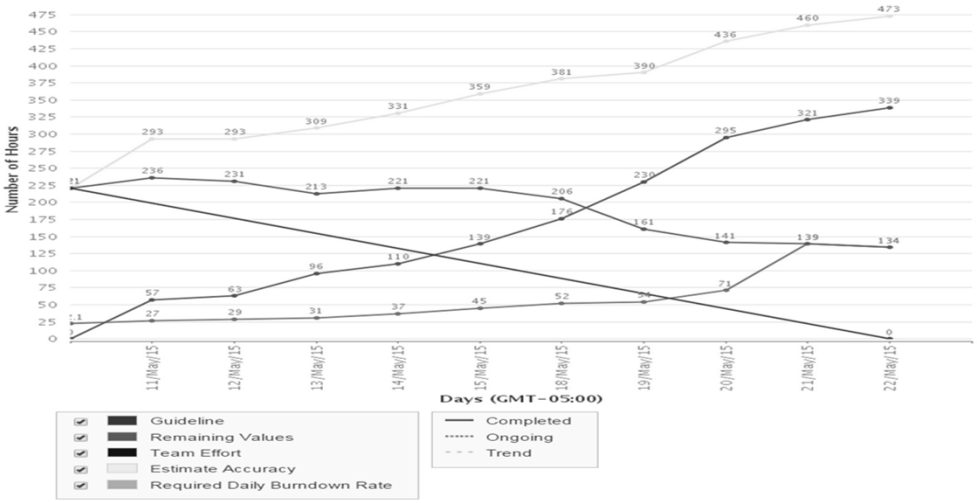


 36

Final Thoughts

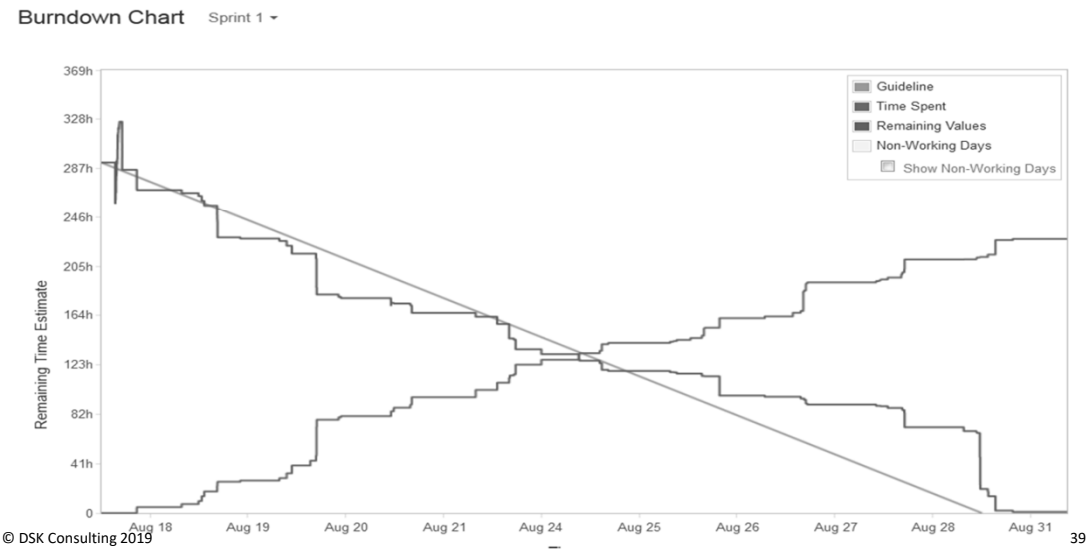
37

Before The Shift Left Transformation



38

Well Into The Transformation



In Summary

- Invest effort to find "bugs" during requirements definition
  - ❑ Develop a qualitative way to measure story quality
  - ❑ Model the execution to gain insight if there are gaps
  - ❑ Introduce the "Product Owner" activities
    - Does it solve a customer problem and add value
    - Can we predictably, deliver to the team's commitment
- Get the testing effort defined, implemented (Test Drive Development)
  - ❑ Ensure you implore different types of testing in your strategy
  - ❑ Make the testing activities transparent
  - ❑ Let the tooling push completion of testing
- Define an Infrastructure that "builds quality in"
  - ❑ Let the system "police" quality
  - ❑ Bring tools to enable others to build in quality (self-serve, transparency)

Move Problem discover and resolution upstream in development, (a.k.a) *To The Left!*

Questions

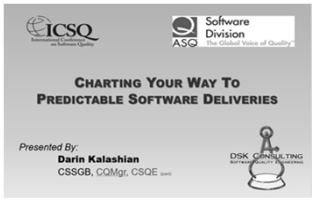
 41

Darin Kalashian

Darin Kalashian has over 30 years of software development experience, with key accomplishments at the organization and individual levels (and everywhere in between). Darin has developed a solid understanding of how to use software methods, tools, and techniques to deliver successful and effective software products. Darin skillfully leverages experience from multiple technology-based business domains that include; DoD/air traffic control & weather detection systems, the public internet (along with the first commercial web browser); automotive audio systems; software-based design, modeling and simulation tools; and enterprise data protection, security, and risk management.

Working within Agile settings, Darin has achieved certification as Professional Scrum Master and Backlog Refinement Practioner. He enjoys coaching teams as they embrace developing product in an iterative, team approach framework. Relying on his past experiences, Darin helps teams set their sights on "win-win" scenarios, where they pro-actively developing high quality software in a timeframe that achieves high customer satisfaction while driving the larger organization's goals and mission. Complementing his agile practices, Darin has a deep foundation in quality tools, were he has been certified by the ASQ as a Certified Software Quality Engineer, Certified Quality Manager, and most recently, as a Six Sigma Green Belt.

Darin's passion is innovating solutions that leverage his quality background and tools, like Six Sigma and Lean analysis, to drive continuous improvements in the software/technology space, most recently, in the DevOps/Continuous Integration and Delivery setting.
Please feel free to reach out to Darin at darin@dskquality.com.



 42