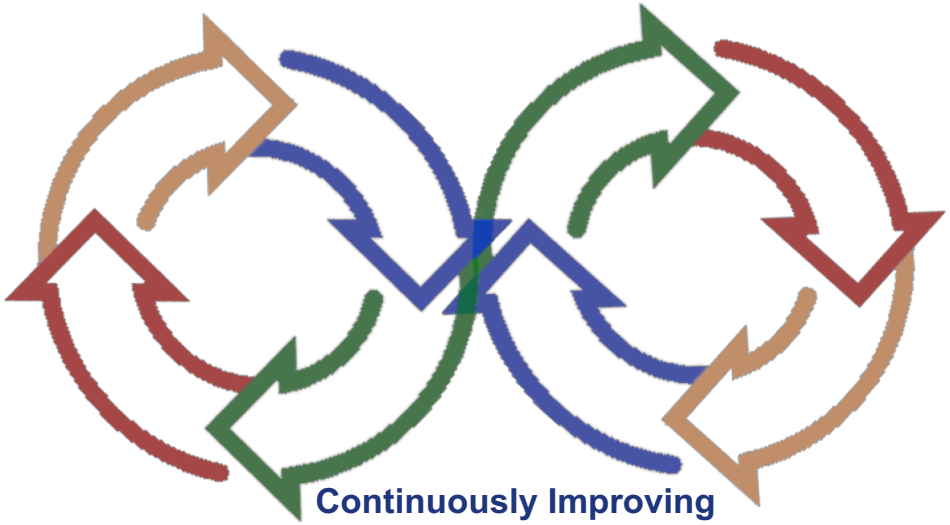
Creating a Culture of Dev Ops Innovation and Agile Practice Change on System z

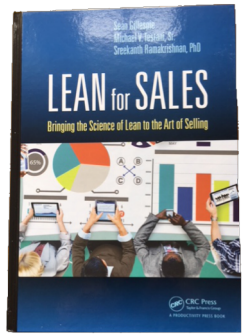
Sean Gillespie
Development Operations Change Leader - Europe

November 2018
Session MJ



Enabling Time For Bright Ideas

Biography



Sean Gillespie is a skilled change consultant with strong facilitation skills and a passion for changing company practices and culture. Sean's engagements utilise the very latest "Visual Thinking and Fun Learning" techniques to engage with your team, and to make change sticky! A staunch advocate of listening to the Voice of the Customer in business. His engagements are iterative and utilise Agile planning to progress change through factual, data-led customer engagements. Sean is the principle author of "Lean for Sales: Bringing the Science of Lean to the Art of Selling". A book published to assist sales teams to use Lean and Agile techniques to improve customer collaboration and engagement.

Today Sean works as a Development Operations Change Leader for Europe at IBM. Part of a global team. Leading collaborative engagements with development and operations teams, as well as C level management to change development operation practices and improve velocity. Sean has successfully delivered engagements at some of Europe's largest banks, retailers and insurance customers. With more than 15 years of experience in working with developer's on IBM System z and distributed systems, Sean's engagements are often focused on modernising practices on System z in line with the broader enterprise development changes.

A native "Aussie" living in the United Kingdom, at IBM he has been selected to receive an Eminence Award, VP Award, and Equity Award for his achievements in delivering value to customers. Sean is a Certified Six Sigma Green Belt and Lean Black Belt.

'Development change is a battle of inches'

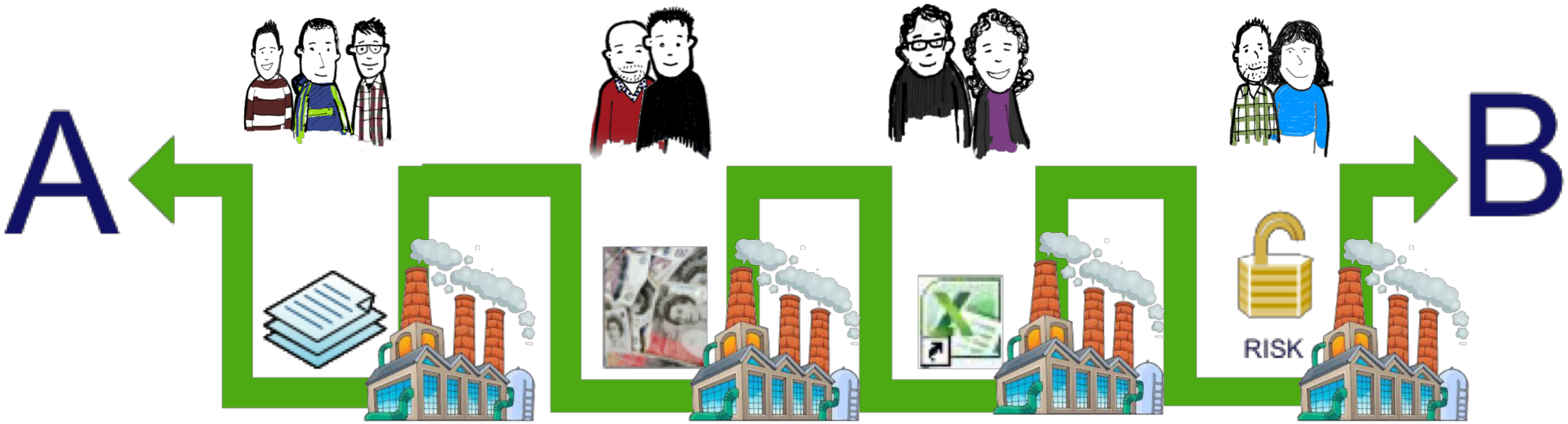
DevOps Workshops In Action

‘Development change is a battle of inches’

Straw Poll

Does Your Development On IBM Z Work ?

Yes or **No**



Hidden Factory

Internal / External Customer

‘Development change is a battle of inches’

Change Has TWO Simple Rules

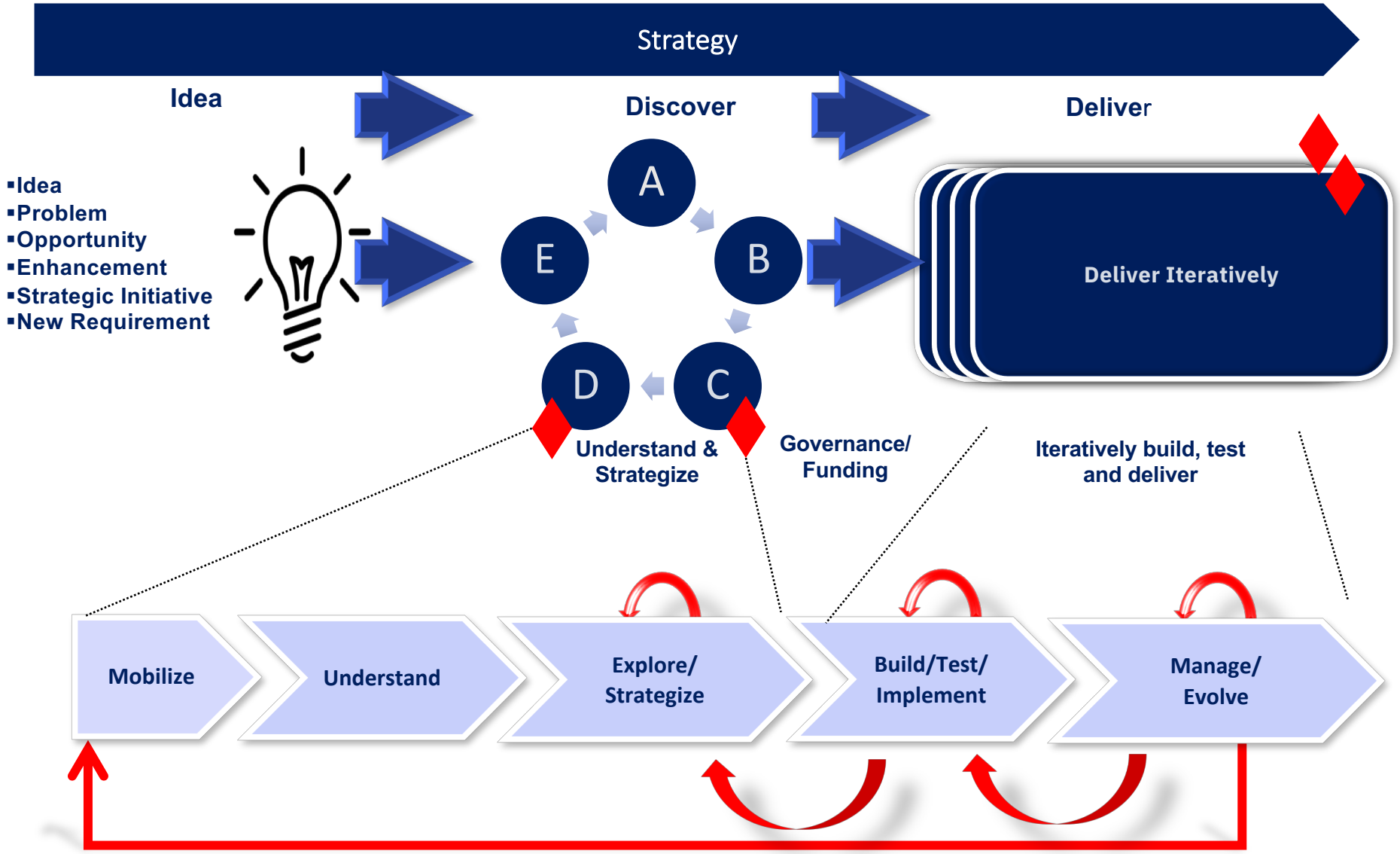
‘Development change is a battle of inches’

Heroic Acts

- Rolling back a production release gone wrong.
- Development team working additional hours to get a project back.
- Having multiple meetings with various team members.
- Spending hours to clarify requirements.
- Returning back to development to fix bugs identified late in development (Systems and Integration Test)

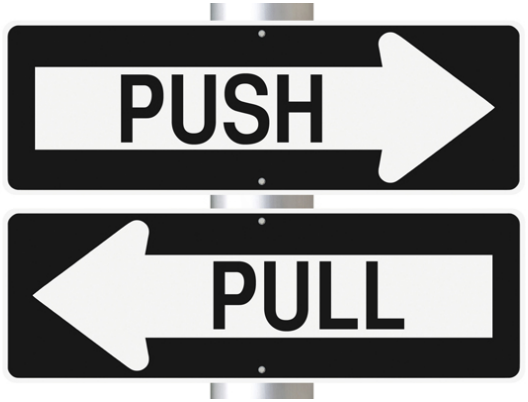
‘Development change is a battle of inches’

SWAT Agile Customer Engagement Model



Agile teams setup to be *intrinsically* motivated

Extrinsic motivation: *An externally applied influence and usually based on a classical approach of rewards and punishments.*



Intrinsic Motivation: *The desire to do things because they are important to us, they matter and because we like doing them.*

Daniel Pink, 'Drive'

Autonomy – the urge to direct our own lives

Mastery – the desire to get better and better at something that matters

Purpose – the yearning to do what we do in the service of something larger than ourselves

Mobilize

Start by Defining The Problem or Goal

GUIDE SHARE EUROPE UK REGION

The mainframe development team is focused on becoming more Agile, continuously improving their development operational practices and moving towards *CI/CD. Our goal is **focused on improving mainframe developer efficiency** and **productivity**, and **reducing mainframe development delivery costs** while improving release practices. A key outcome of this goal is to improve speed to market in line with market demands, whilst reducing costs. We will achieve this goal by a shift left in code quality checks and testing, implementing independent mainframe development and testing, and eliminating waste (Motion, Defects, Waiting etc). Our team are focused on achieving this goal by end of Q2 2017.

Hill 1 - Test / QA

A mainframe developer can get a runnable set of new repeatable test cases for a large module automatically within 60 minutes

Mainframe development teams are determined to become more Agile and to continuously improve their development operational practices. Our goal is focused on improving developer efficiency and productivity, cost effectively delivering releases covering: a major release every other month, interim releases every even month, and minor changes weekly unless preceding a monthly release. A key outcome of this goal is to improve Speed to Market in line with market demands, whilst reducing operational costs. We will achieve this goal by becoming more Agile, implementing **Automated Developmental Practices, Eliminating Waste (Motion, Defects, Waiting) and Improving Delivery Reliability and Efficiency.** The mainframe teams have already made progress on this journey and will continue to implement changes towards the achievement of this goal.

Hill 5 - Test / QA

A mainframe developer can determine the most relevant test cases to run based on a code change in less than 5 minutes.

Agree Personas to Identify our Target Audience

Nitin
Vendor Mainframe Developer (offshore)
10 years experience in development, of which 2 years within <Insert Your Logo>
COBOL, DB2
35 years old, married with 3 kids
Lives in Mumbai

Robert
Vendor Mainframe Developer (Onshore)
25 years experience in development, of which 25 years within <Insert Your Logo>
COBOL, DB2
55 years old, married with no kids
Lives in London

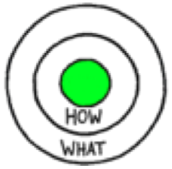
Bert
Vendor Mainframe Developer (Offshore)
3 years experience in development, of which 2 years within <Insert Your Logo>
COBOL, DB2
25 years old, married with no kids
Lives in Copenhagen

Preeti
Vendor Mainframe Tester (onshore)
6 years experience in development/testing, of which 5 years in <Insert Your Logo >
COBOL, DB2, 27 years old, married with 2 kids
Lives in Zurich

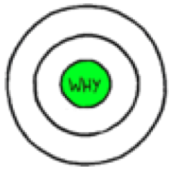
WHICH CAME FIRST

OR THE

Operations and Architecture



Development Practices

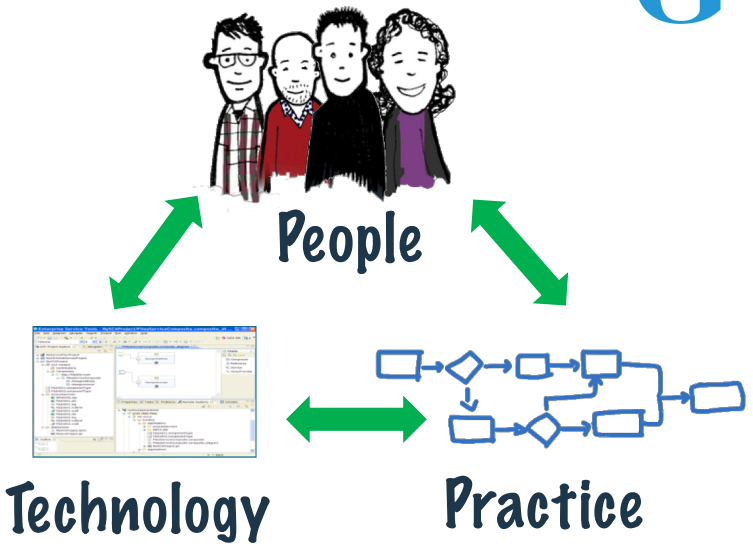


Change without Mandate Means No Change at All

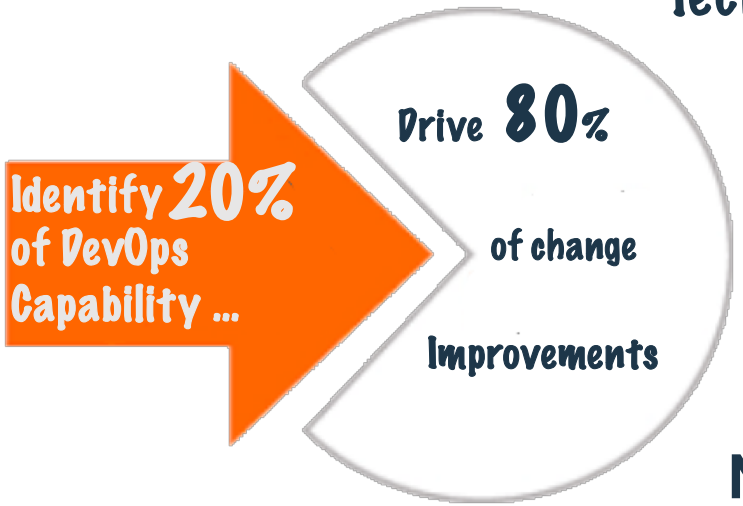
70% RIGHT

Not 100% WRONG!

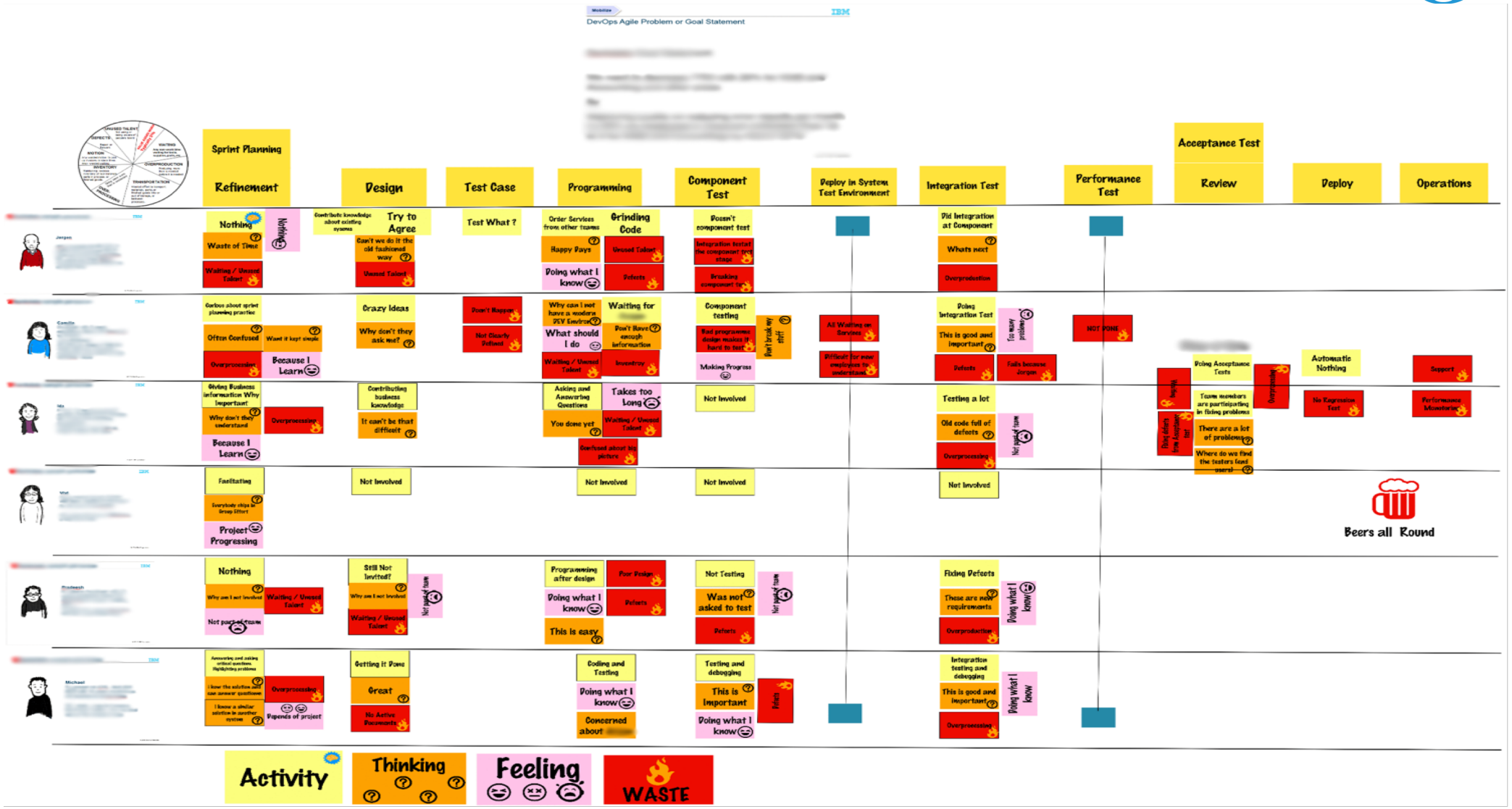
Core Principles of Change



Intrinsically Motivated Teams of Developers

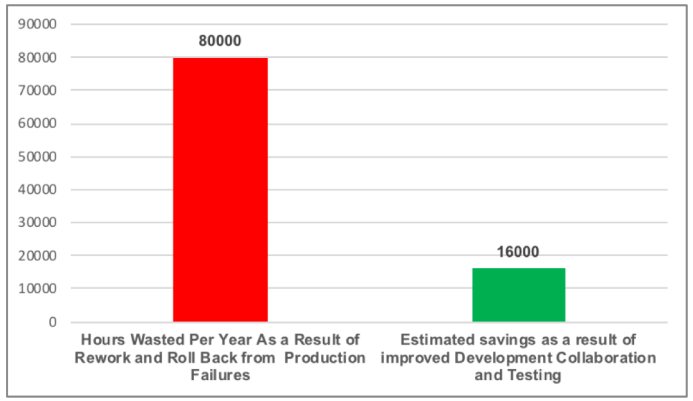


What's Your Development Story

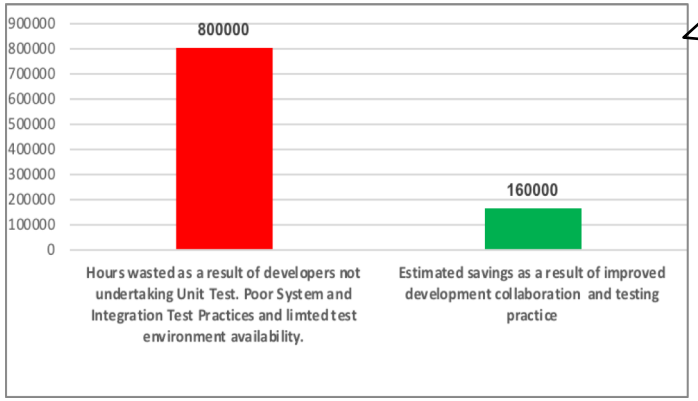


What's Your Development Story

Consequences of Waste

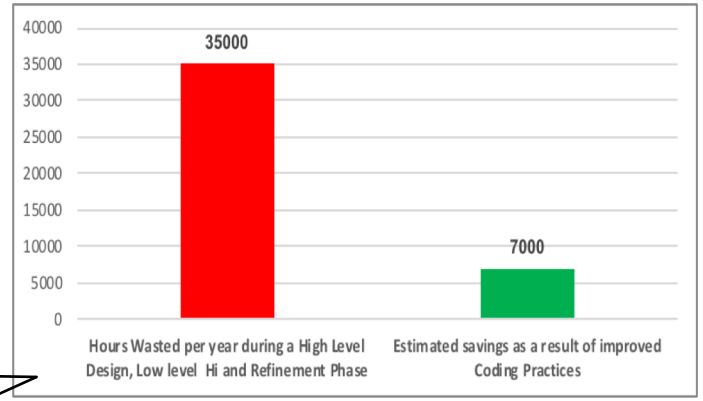


Hours Wasted Per Year As a Result of Rework and Roll Back from Production Failures	Estimated savings as a result of improved Development Collaboration and Testing
80000	16000

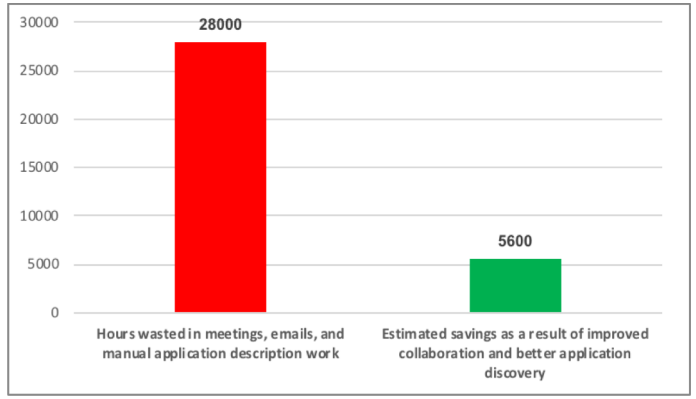


Hours wasted as a result of developers not undertaking Unit Test. Poor System and Integration Test Practices and limted test environment availability.	Estimated savings as a result of improved development collaboration and testing practice
800000	160000

Target 80% Reduction in Waste



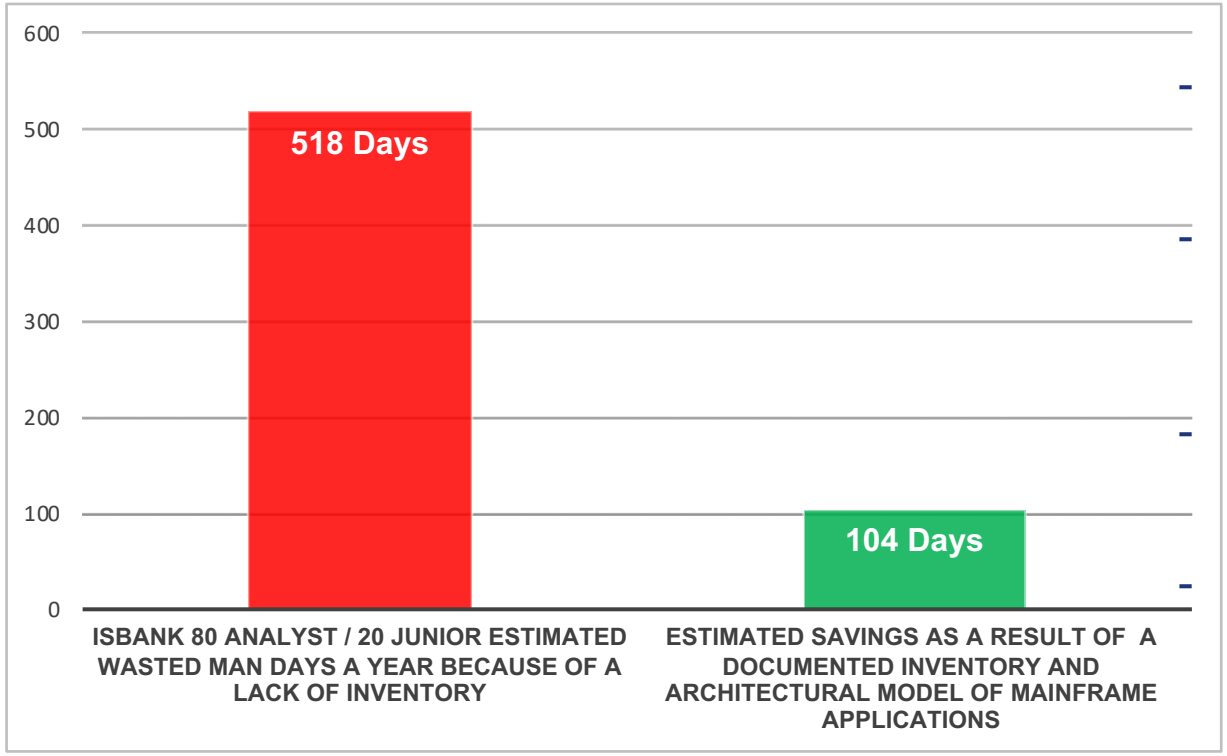
Hours Wasted per year during a High Level Design, Low level Hi and Refinement Phase	Estimated savings as a result of improved Coding Practices
35000	7000



Hours wasted in meetings, emails, and manual application description work	Estimated savings as a result of improved collaboration and better application discovery
28000	5600

Consequences of Waste

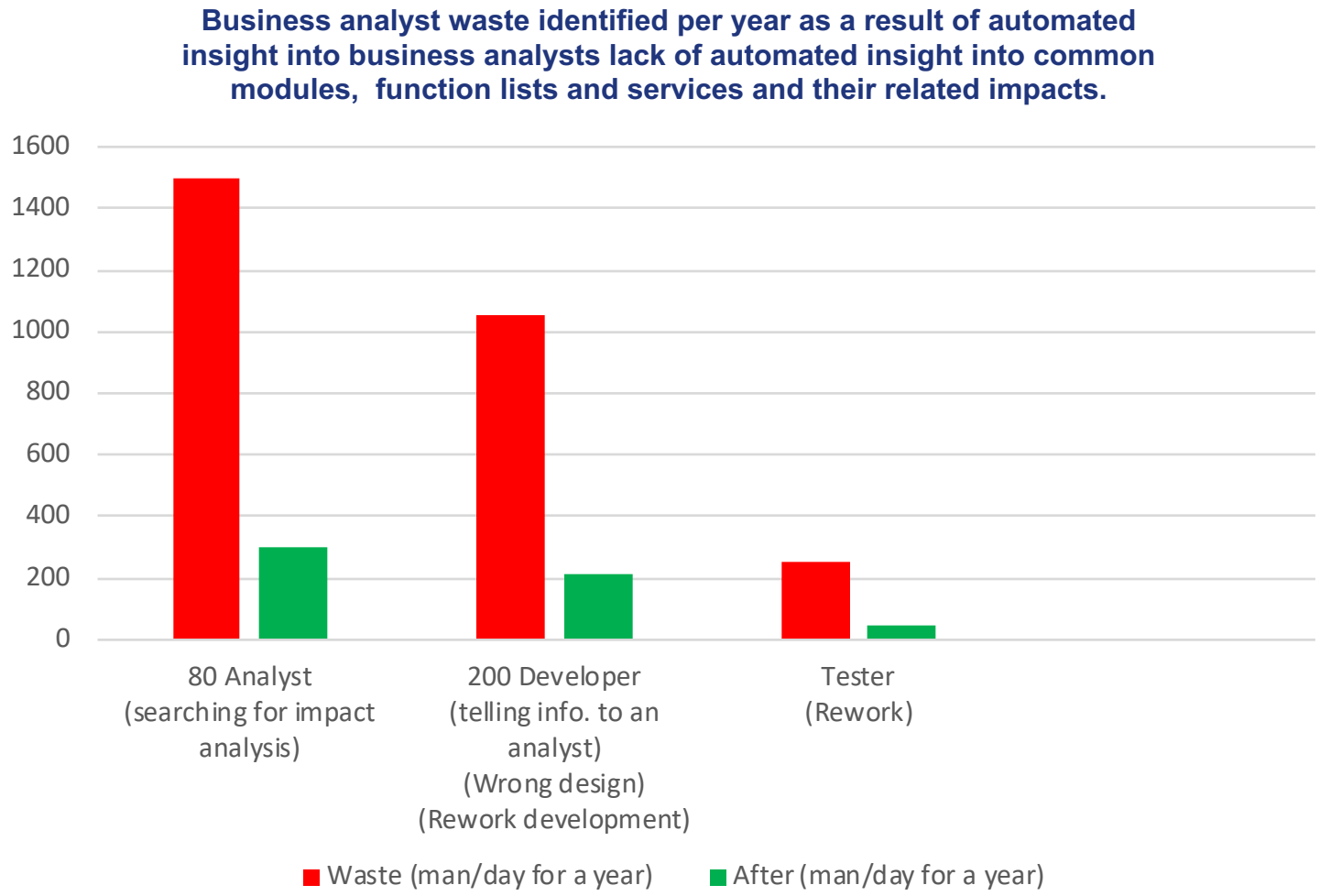
Change and Release Waste Conclusions



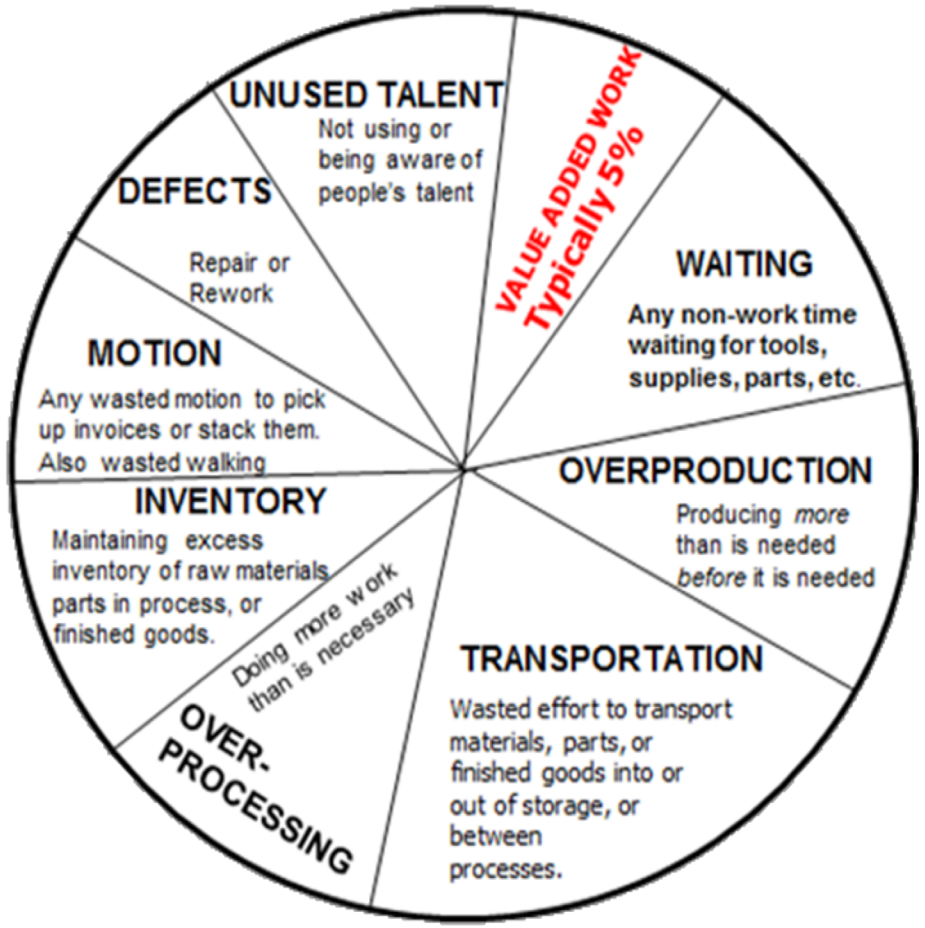
Test Automation Waste Conclusions

- ***9600 Defects***: Resolving per year on Mainframe
- ***1200 Defects***: Defects reopened and redeployed
- ***3 hours***: Average deployment waiting time to UAT
- ***3240 md***: Total waste time per year
- ***240 people***: 200 developers, 40 testers affected
- ***13,5 days***: Waste time per person, per year

Consequences of Waste



Categorize and Quantify Wastes for Removal



Defects: A higher than average number of defects found late in the development, test and production stages. The increased defect rates are as a result of a heavy dependence on an individual developer's capability and planning and the silo'd alignment of development teams and skills. Based on industry experience, defects found late in the cycle can cost up to 10 x more to resolve, than those identified early in the cycle.

Over-processing: Development practices are often manually based. This requires individual developers to undertake increased work throughout the SDLC lifecycle. A good example that was discussed during the workshops was the constant creation and modification of data for testing. Additionally the large number of meetings are undertaken during the design analysis phases. Over-processing increases operating costs, hampers staff from moving to another project, and impacts project delivery time lines.

Over-production: Current development cycle requires a developer to produce more than is actually needed of required by the customer. An example of Over-production is the creation of a test plans or documentation that is never used or reused. Over-processing increases operating costs, hampers staff from moving to another project, and impacts project delivery time line.

Motion: Development practices require increased levels interactions between team members to progress a project or task. Requests can often result in multiple manual exchanges before the correct skill or information is identified. A high level of motion waste exists during the Requirement Analysis and Development phases.

Waiting: Development practices are impacted by waiting. Waiting for a user acceptance test environment and data to be available.

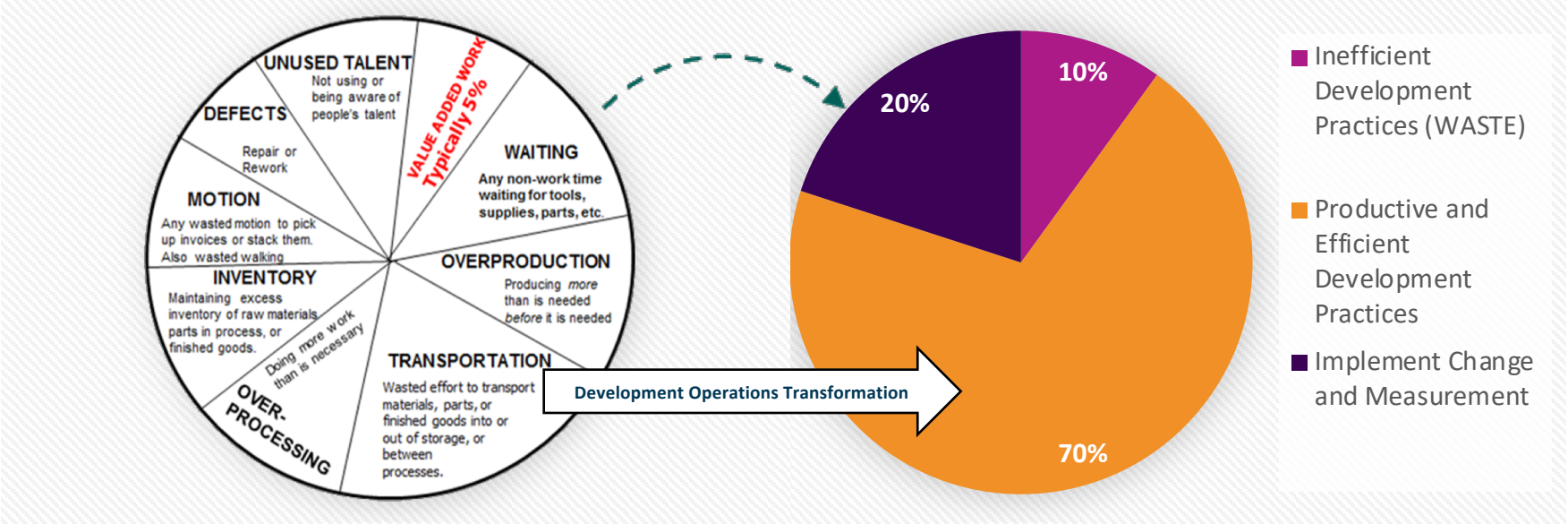
Unused Talent: Probably the most destructive and costly waste. Modern banks are very lean with SME skills. Wastes like over-processing, Motion and Defects, inhibit SME's from focusing their effort effectively on value added and personally rewarding activities as opposed to value draining activities.

Explore/ Strategize

True Cost of Application Delivery Lifecycle

GUIDE SHARE EUROPE UK REGION

Daily Development Activity and Collaboration



Hidden Factory= Additional value that can be created if Wasteful Development Practices are eliminated and redirected those resources to Innovation & Delivery Throughput

Iteration 1: Agile User Stories To Remove Waste

User Story 1

As a mainframe developer I want a tool that shows me the call hierarchy and code overview structure so that I have a better overview and it is easier to make changes to existing code.

How do I get <meta data> about my system?
Meta data can be:
Interdependencies between programs
Databases
Data structures
SI components.

User Story 2

As a mainframe developer, I need a framework for creating, executing and managing automated test to increase efficiency and productivity

User Story 3

As a developer, I want a standard practice for unit test case development and code coverage capability so that I can continuously test for quality and defect removal

Criteria:

Create a standard test framework

User Story 4

As a developer, I want to have defect tracking mechanism so that I can assess test quality and have defect reduction in future releases

Criteria:

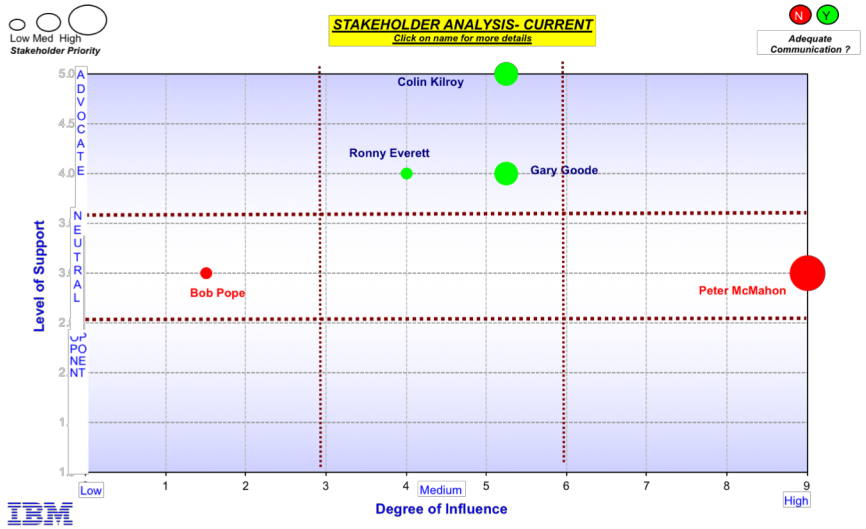
Create a central repository for defect tracking

Iteration 1: Further Confirm the Problem

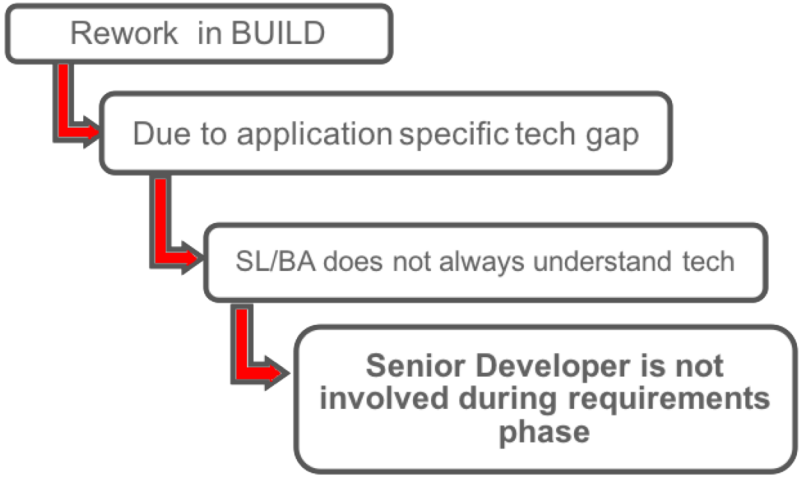
Results Inter-dependancy

Sample size	100	100	100
Time per day	10min	35 min	45 min

Change Project Stakeholders



5 Why's Root Cause Analysis



Iteration 1: Build a Minimal Viable Product

Identify Module
Select the Module Which Needs Change

Isolate Code
Isolate the code from all dependencies

Input Dummy data
Replace external calls with Dummy Data

Select Input / Output
In Watch select I/O from Variables

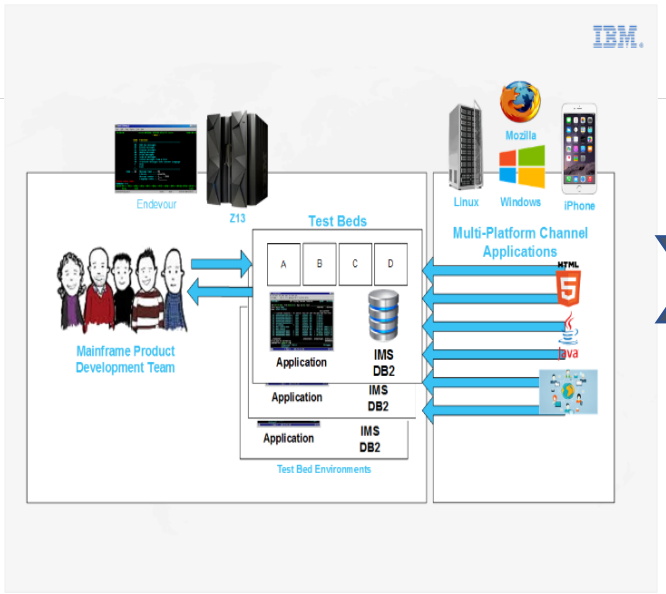
Copy/Input Test Data
Enter values for Input Variables

Execute the Test Case
Run the Test Case

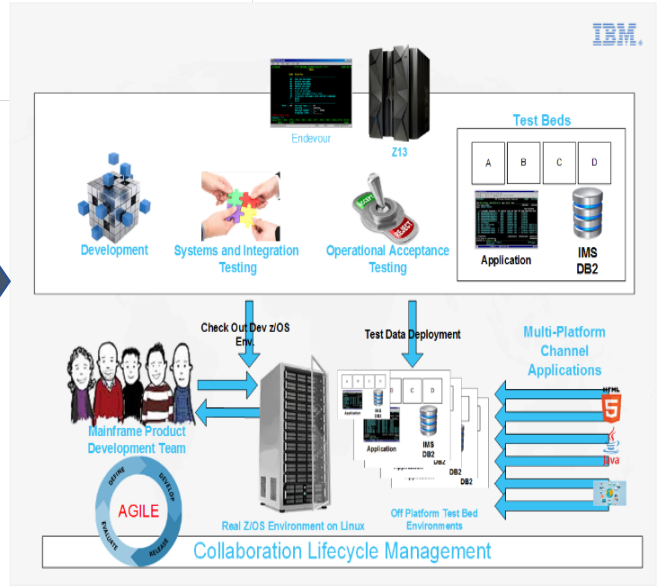
Auto Documentation of Test Results
Store the Test Case Automatically

Generate Report
Code Coverage, Test Coverage and Code Quality

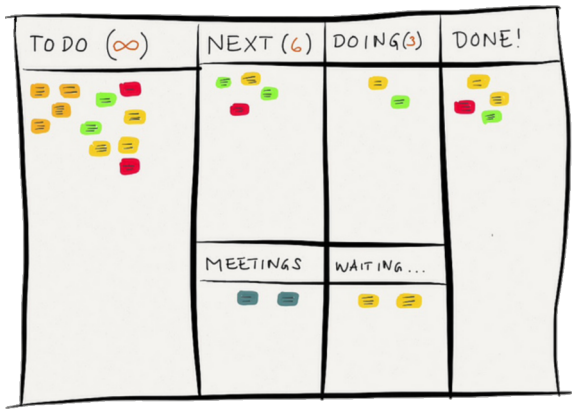
Current Practice



New Practice



Iteration 1: Build a 30 Day KANBAN Board



Build 30 Iterative Plan for Change -

- With a small agile intrinsically motivated team build a Kanban to eliminate waste and change the world.
- Allocate as little as 2 hours a week for 30 days
- Further confirm the size and validity of the waste
- Confirm the root causes associated creating the identified problems
- Agree standards i.e. What is a Unit Test
- Architect a new solution and Plan for implementation.
- Feed back to Senior Management the Results
- Build an Agile Communication Plan focused on Communications Channels, Engagement Test Approaches, Collaborative Culture, Train and Learn

Agile Customer Engagement – Iteration 1

People

Process / Practice

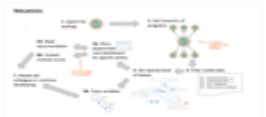
Technology

Mobilise a Team

Understand and Explore

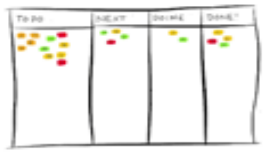
Intrinsically Motivate to Lead Change

Build Agile Storyboard focused on Waste



Build a MVP new practice

Focus on Waste that inhibits development practices and innovation



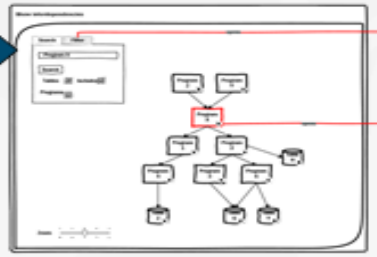
Build plan to confirm wastes, test MVP, seek consensus as to value

Use Lean and Agile tools to Identify ROOT Causes

Team Stand Ups and Product Management meetings to review Progress and Course Correct as Required

Manage and Evolve
Adjust approach and change according community need

Build and Test



Propose MVP and the individual benefits to the developers.
Management support and drive for change
Educate development community 'HOW' change is extrinsically valuable to them personally.

Create pull because community see value

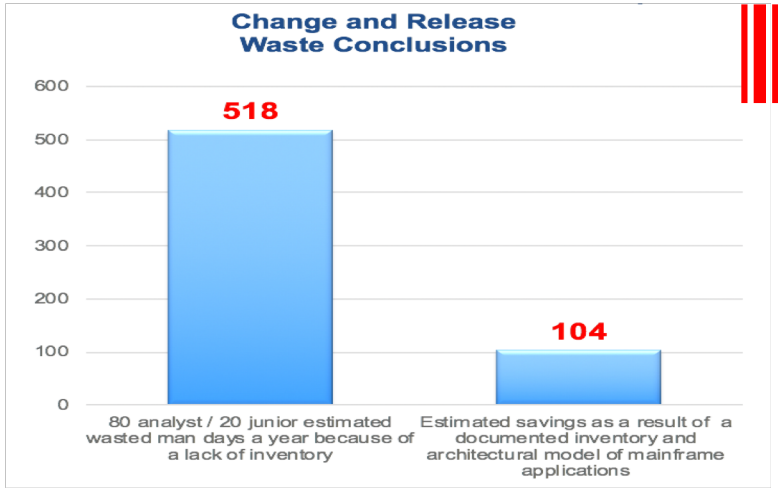
600 strong development community

Workshop Outcomes

GUIDE SHARE EUROPE UK REGION

Day 1

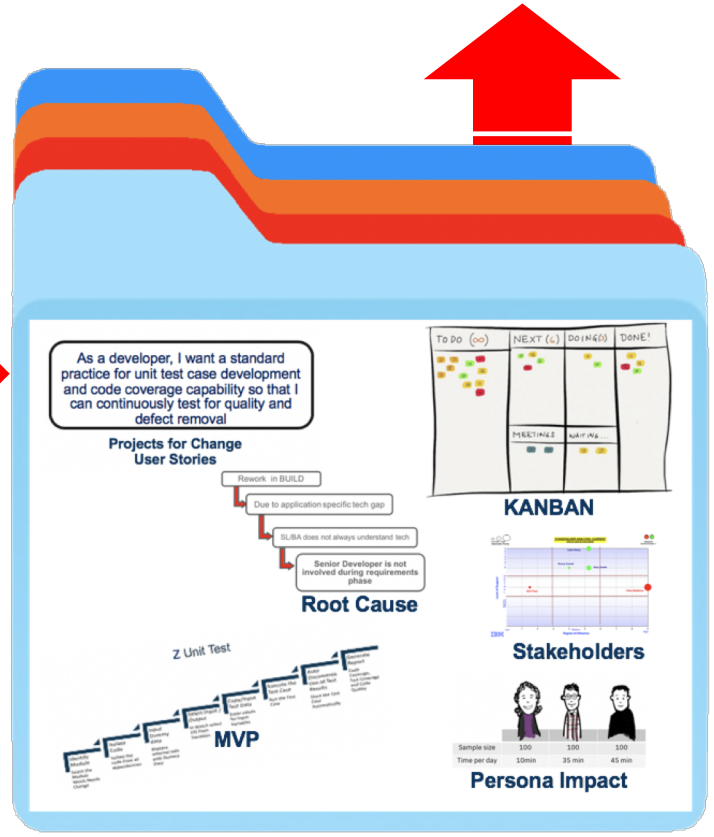
Point in Time View of Practices, Waste and Where to Start



Quantification of Waste

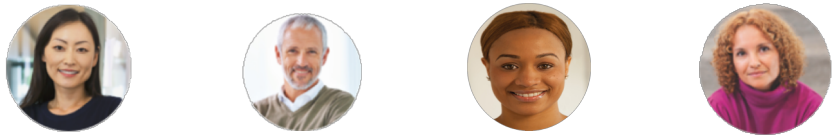
Day 2

Team Presentations to Senior Management



Now have up to 6 projects to begin Dev/OPS change

Workshop Iteration Outcomes

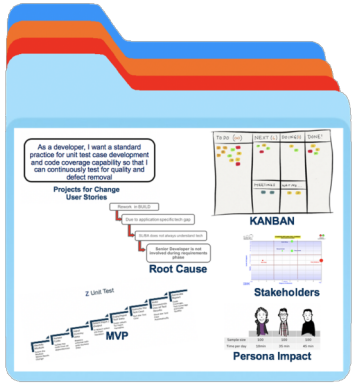


Management Updates and Sponsorships

Part Time Project

Iteration 1-3 (2 hours a week for 30 Days)	Outcomes
Test Waste and Root Cause assumptions Confirm and engage stakeholders Test and Improve Minimal Viable Product (MVP) Agile Stand Ups and Product Management Meetings	1. Confirmation of waste cost and size 2. Further clarification and testing of proposed solution 3. More detailed KANBAN
Improve MVP expand and align Operation and Architecture. Modify MVP as required Build Kanban to define functional deliverables Continue to test with an expanded set of interested peers Agile Stand Ups and Product Management Meeting	1. MVP-functional deliverables defined and agreed with stakeholders 2. Backlog being built with greater clarity on timeline 3. Management support for timeline and value delivered
MVP broken down to backlog of work with user story deliverables. Projects prioritised and with Senior Management Support Agile communication plan built Kanban build to point it can become a funded project.	1. Backlog of work defined 2. Functional deliverables defined and agreed. 3. Confirmation of budget formalised project 4. Clearly defined team to implement. Team as time to deliver. 5. Agile Communication Plan

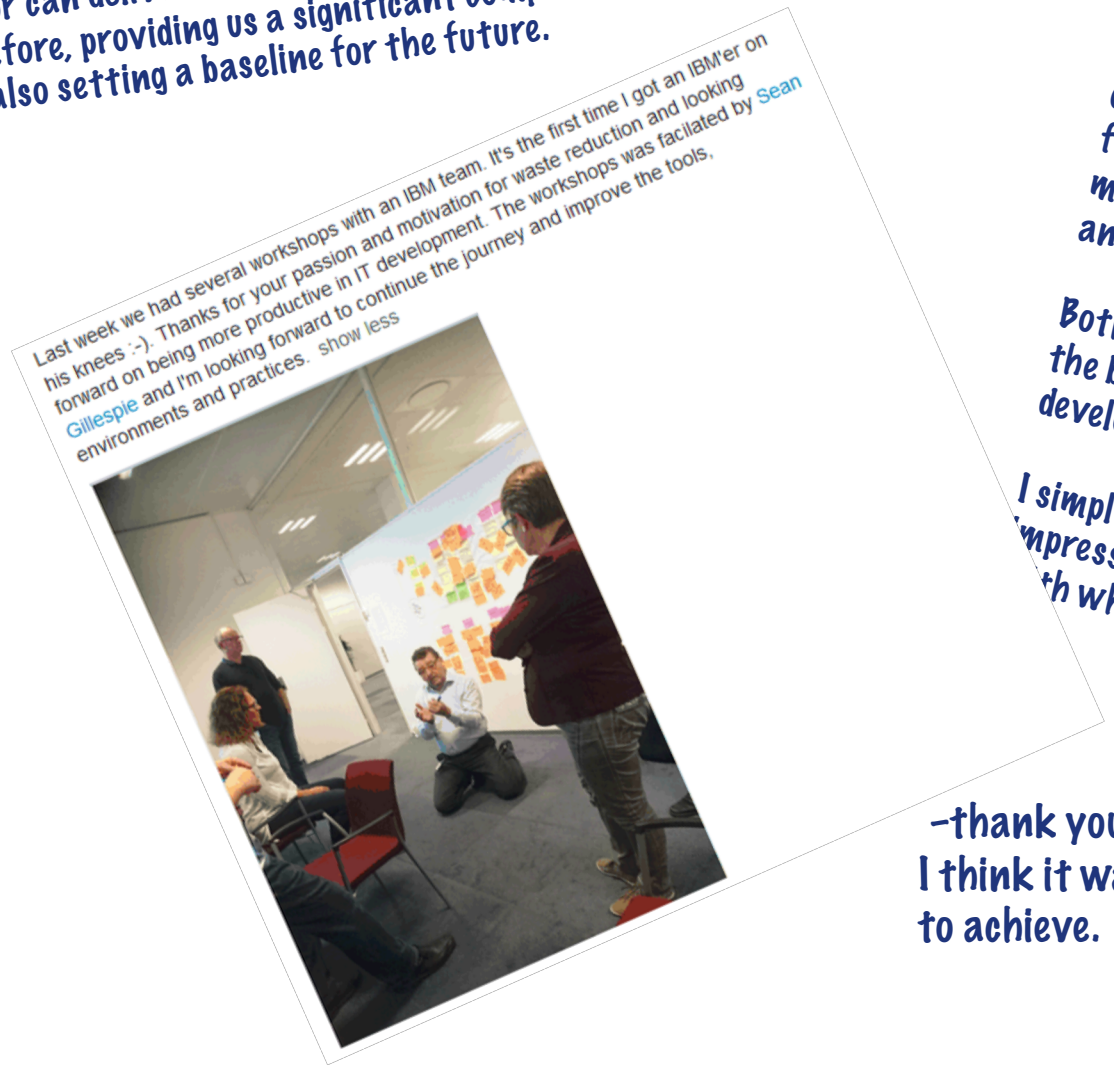
Funded Project



Full Time Project
Defined Budget
Sprint Team
Backlog
Function
Deliverables
Agile
Communication Plan

Customer Feedback

I want to thank you for the excellent work done. For the local IBM team as well as for our customer the workshop was an eyeopener. All of us are convinced that no other vendor can deliver this kind of customer involvement. So therefore, providing us a significant competitive advantage and also setting a baseline for the future.



At Company A, , we have now, together with IBM driven by Sean Gillespie, conducted two very valuable workshops, one autumn 2017 and very recent one in Bangalore, both workshops were based on creating value through identification of waste and how to accommodate and ultimately get rid of the same. Attached you will find the combined result from the latter, where actual proposals were made by the teams on how to reduce waste and work smarter, presented for and accepted by management at the same time. This means that we have now very good basis for developing and implement the individual proposals.

Both workshops have also provided very useful input to the business case of implementing complete new development model and environment for mainfram.

I simply need to let you know that I have been very mpressed by the skill, great effort and collaboration h which this work has been done.

-thank you for a great effort done by you all.
I think it was spot on compared to what we wanted to achieve.

We want your feedback!

- Please submit your feedback online at ….
  - ➢http://conferences.gse.org.uk/2018/feedback/mj

- Paper feedback forms are also available from the Chair person

- This session is MJ