Digital Assurance

Driving Continuous Adaptive Testing Practices

Paul Gerrard
Gerrard Consulting

Jonathon Wright
Director of Digital Assurance

Disclaimer

Certain information in this presentation may outline CA's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. This presentation is based on current information and resource allocations as of March 1st and **is subject to change or withdrawal by CA at any time without notice**. **The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion**.

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to CA maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. CA assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. **In n**o event will CA be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages.

Abstract

Paul Gerrard & Jonathon Wright

CA
Director of Digital Assurance

User expectations are shifting faster than ever and yet are set higher than ever before. The software landscape is therefore evolving rapidly to deliver higher quality software at an ever-greater pace. Testing must be able to keep up and move away from a traditional Core IT approach. Shifting to an Adaptive IT model calls for a new practice that requires accelerated communication, collaboration, integration, measurement and automation.

Whether you label this as Digital Transformation or not, understanding the detail of this journey is an essential part of every organisation's journey to becoming a Digital Enterprise.

Together with Paul Gerrard, Jonathon Wright will define what Digital Transformation is, its complexity and its risks, what Digital Assurance is from the process angle and then they'll look at how models are being at the heart of testing, creating an essential 'blueprint' for test design.

Agenda

1. INTRODUCTION
2. WHAT IS DIGITAL ASSURANCE
3. PROCESS PATTERNS, NEW MODEL FOR TESTING, SHIFT LEFT & RIGHT
4. COGNITIVE ADAPTIVE TESTING
5. WHAT NEXT?
6. Q & A

Digital Assurance - ‘Evolution, over Revolution’

78% of enterprises believe that they need value-driven algorithmic business models to respond to disruption in the next 3 years.

VALUE-DRIVEN

Legacy is your Legacy (API)

Digitalization of your Core (SOA)

Digital (Cognitive Adoption)

MAINFRAME — 1960s

ERP/CRM SYSTEMS — 1990s

APPLICATION ECONOMY (API) — TODAY

TIME

1 – “The Battle for Competitive Advantage in the App Economy”, Oxford Economics, 2015

ca technologies

Journey to Enterprise Digital

UNPRECEDENTED

94%

of executives face increased pressure to release apps more quickly

2014 Vanson Bourne study commissioned by CA

"**Businesses no longer have the luxury of time.** Rising customer expectations, competitive threats, and increased consumer choice have turned rapid delivery and iteration of software applications into a competitive differentiator.

Adoption of Value-Driven Delivery has become the new means to better business outcomes."

TechRadar™: Continuous Software Delivery, Q2 2015, Forrester Research, May 1, 2015

Digital Assurance

Delivering Quality at Speed

VELOCITY

94% of executives face increased pressure to release apps more quickly[1]

QUALITY

2/3 of business leaders say the future of their business depends on the quality of their software[2]

LOWER COST

25% of a single application's development and operations costs is wasteful[3]

While Ensuring a Superior Digital Experience!

1. 2014 Vanson Bourne study commissioned by CA
2. "Surviving Disruption, Leading Change: Winning in the Application Economy," 2015
3. *"DevOps and the Cost of Downtime" – IDC 2014*

"Surviving Disruption, Leading Change: Winning in the Application Economy," 2015

Digital Assurance

Shift Left introducing DesignOps

DesignOps

Create and deliver Ideas to Outcomes

Agile Management	Continuous Delivery	Agile Operations
Redefine how work is planned, managed and executed	Accelerate and streamline development, testing and release	Provide a flawless app experience optimized for performance
Continuous Development	**Continuous Adaptive Testing**	**Continuous Release**

Pervasive Security

ca technologies

Digital Assurance

Continuous Adaptive Testing (CAT)

CONTINUOUS ADAPTIVE TESTING

The ability to reliably release high-quality solutions at any time

Accelerate Deliveries

Reduce Errors

Manage Complexity

Increase Visibility

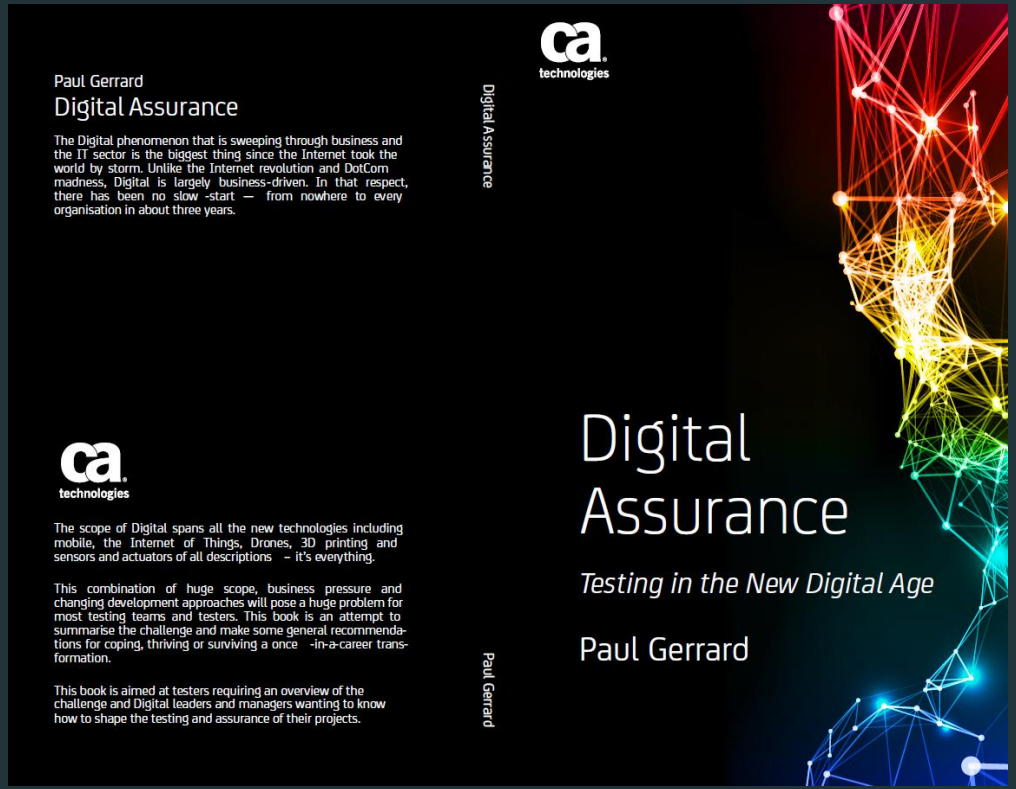
Drive Collaboration

Continually Improve

ca technologies

Digital Assurance

Pocketbook



Paul Gerrard

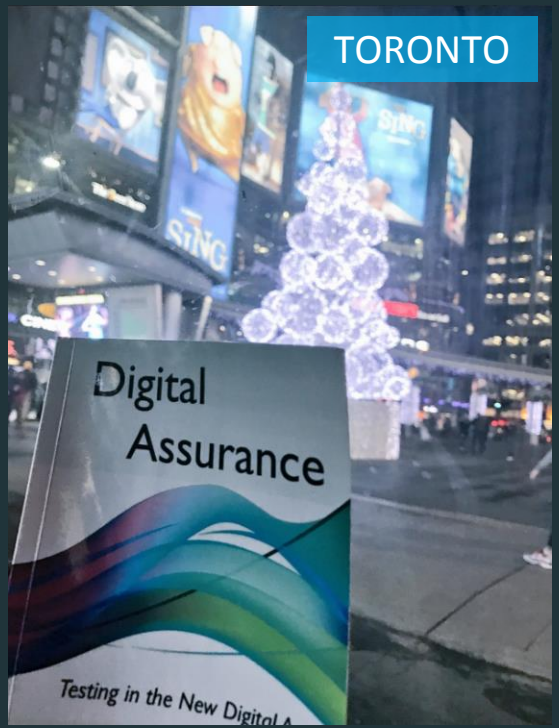
NEW exclusive Pocketbook authored by Paul Gerrard, eminent Test-Transformation consultant, thought leader, teacher and international conference speaker.

Order your FREE copy today from http://bit.ly/digital-assurance

ca technologies

Digital Assurance

Pocketbook – World Tour



ca technologies

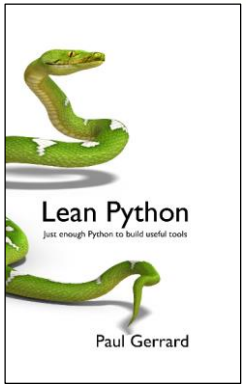
Notes on Digital Assurance

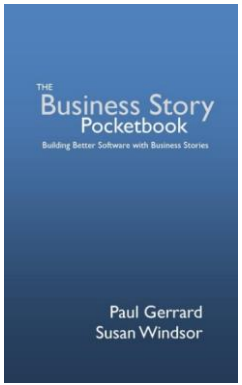
Process patterns, new model, shift-left, shift-right and tools

Paul Gerrard

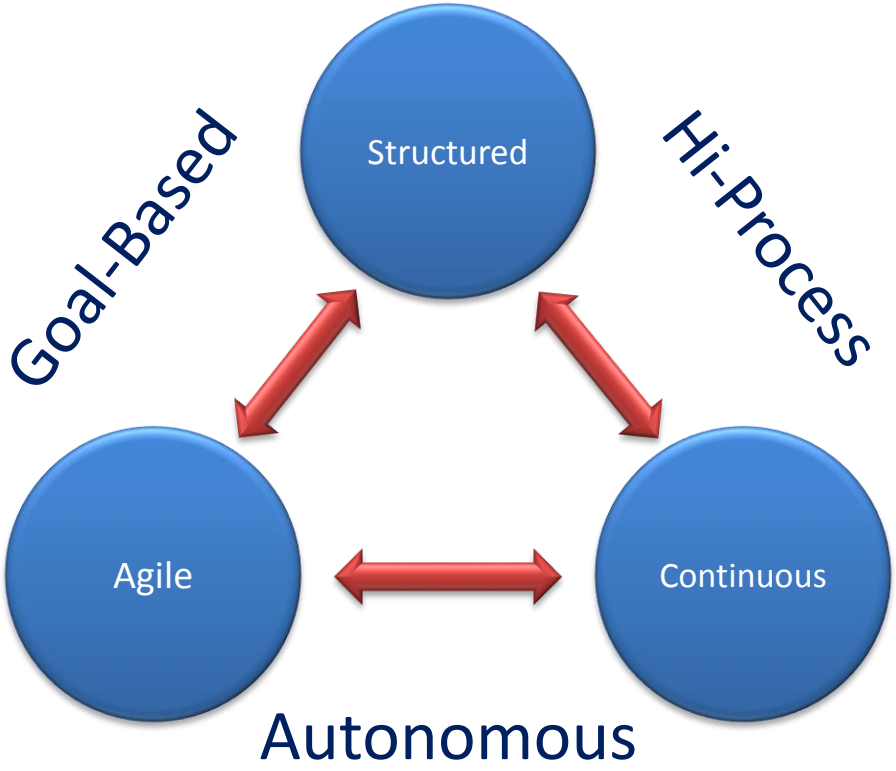
paul@gerrardconsulting.com

gerrard consulting





Three development patterns



Characteristics of the patterns

Characteristic	Summary
Structure	What is the organisational structure of the project team?
Pace/cadence	What drives the rate of decision making? Who do decisions depend on?
Leadership	How is the team managed/directed? What style of leadership is involved?
Definition	How is requirements knowledge captured? In what format?
Testing	How is testing (mostly) performed? Scripted, exploratory, automated?
Automation	When is automation used? Who leads the automation effort?
Measurement	What/how is project measurement performed?
Governance	What form does governance take?

Profiles of the three patterns

Characteristic	Structured	Agile	Continuous
Structure	Managed team	Autonomous	Production Cell
Pace/cadence	Business decision	Team decision	Feedback
Leadership	Project Managed	Guided Research	Line Managed
Definition	Fixed spec	Dynamic spec	Executable Specs
Testing	Scripted	Exploratory	Automated
Automation	Retrospective	Developer led	Pervasive
Measurement	Pervasive	Avoided	Analytics
Governance	Bureaucratic	Trust-based	Insight-Driven

Not three patterns; There are many

You have to work out your own hybrid approach that suits your organisation

The old way of testing won't work in the future

We need a New Model of Testing (free from logistics)

Forget Logistics

(for the time being)

Document or not?

Automated or manual?

Agile v waterfall?

This business or that business?

This technology v that technology?

ALL Testing is Exploratory

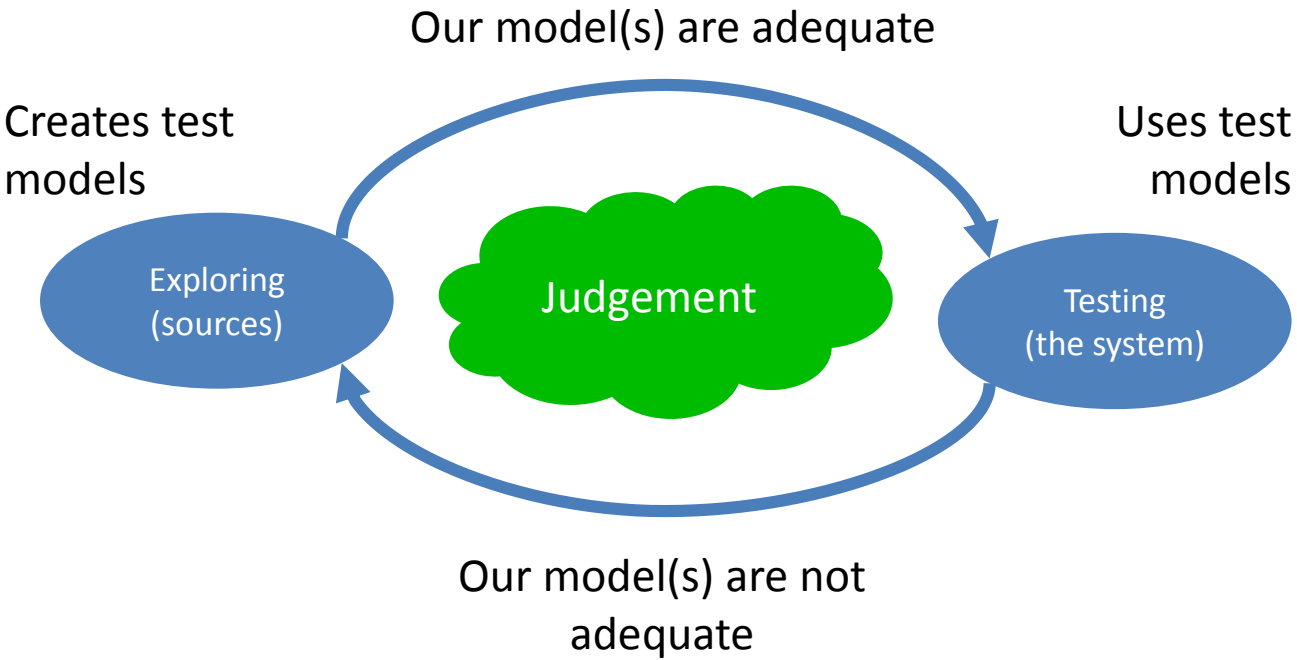
We explore sources of knowledge ...

... to build test models ...

... that inform our testing.

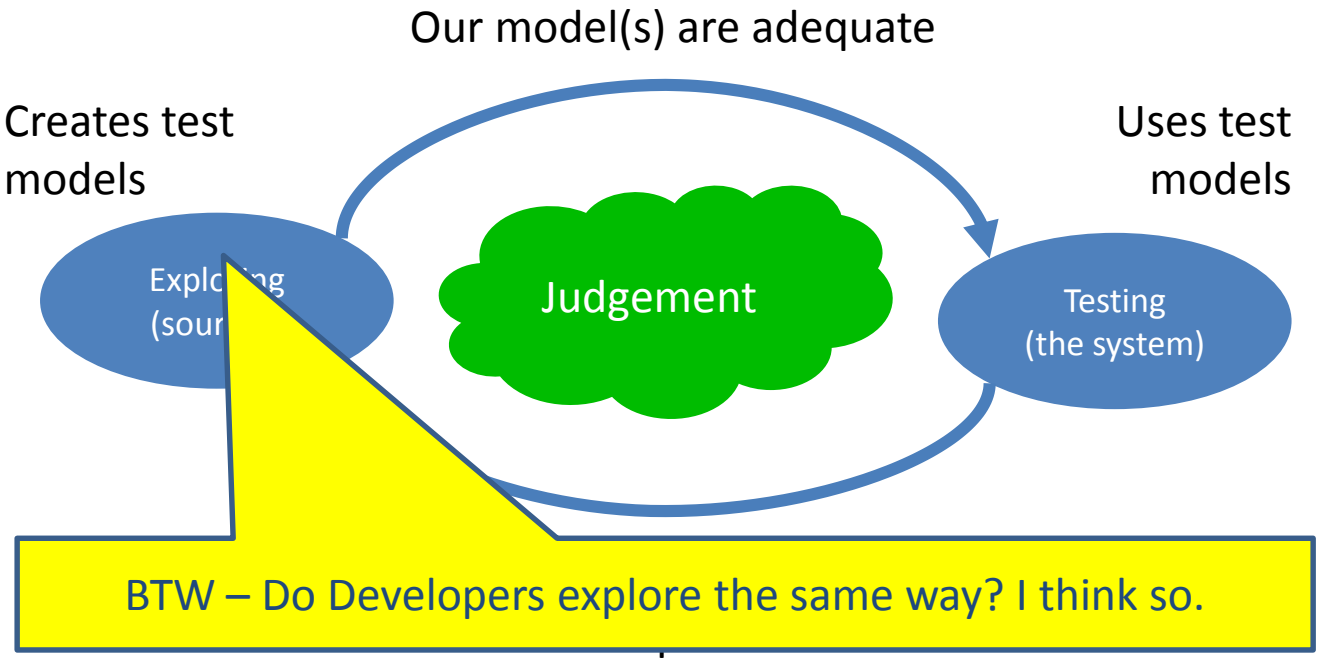
Judgement, exploring and testing

We explore sources of knowledge to build test models that inform our testing

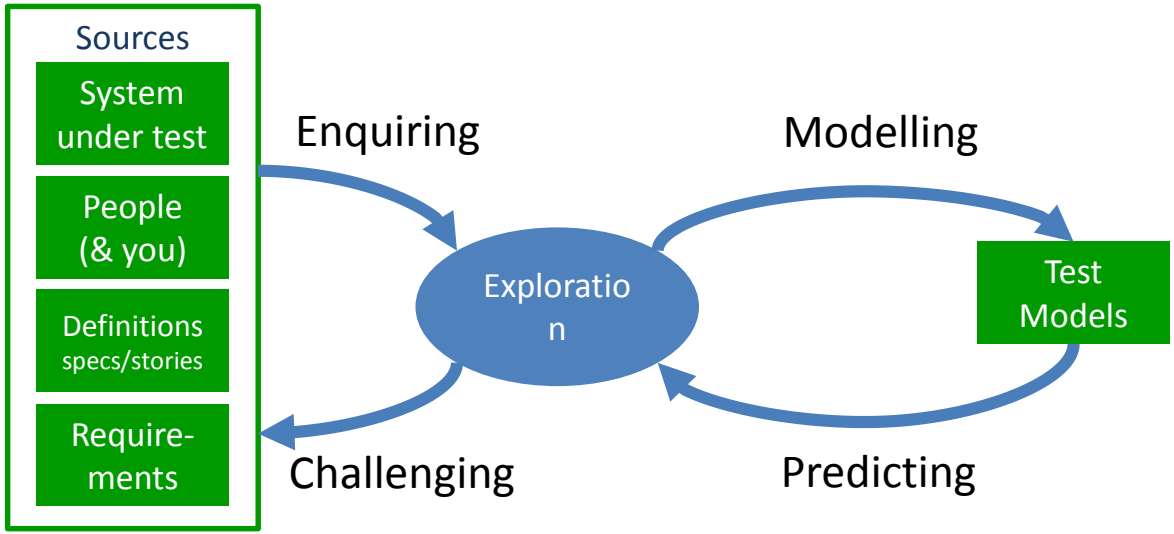


Judgement, exploring and testing

We explore sources of knowledge to build test models that inform our testing



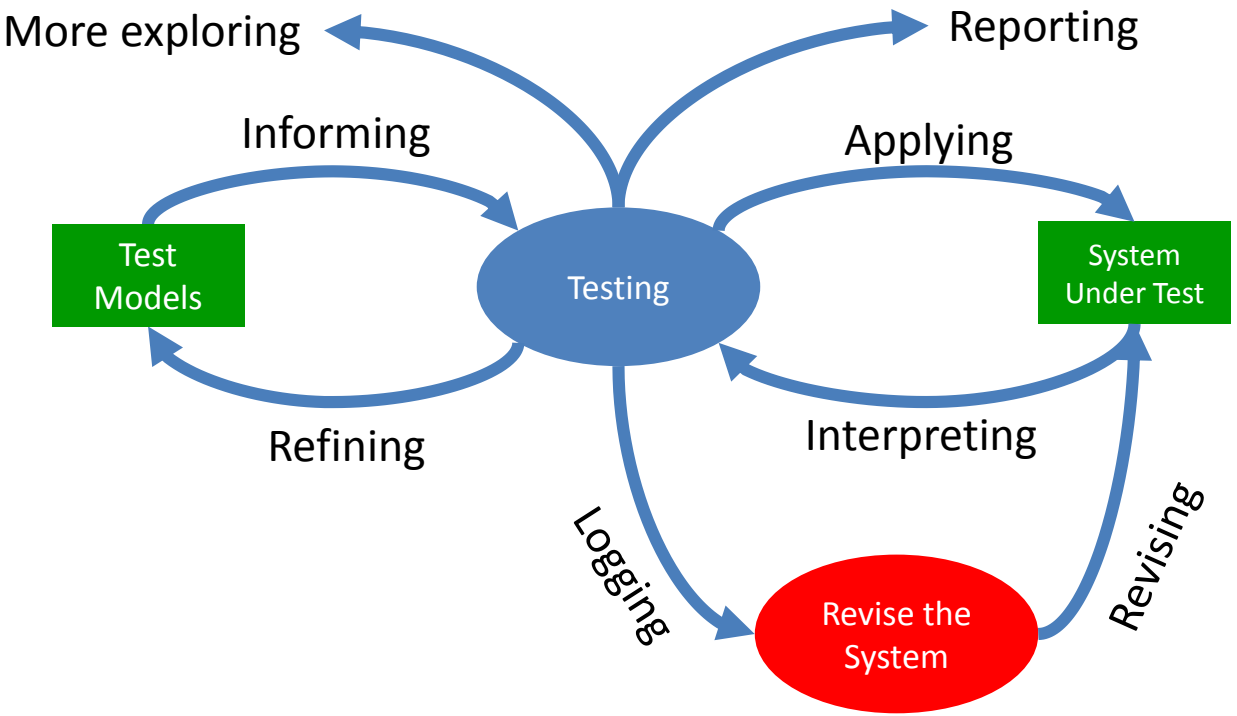
Exploration process



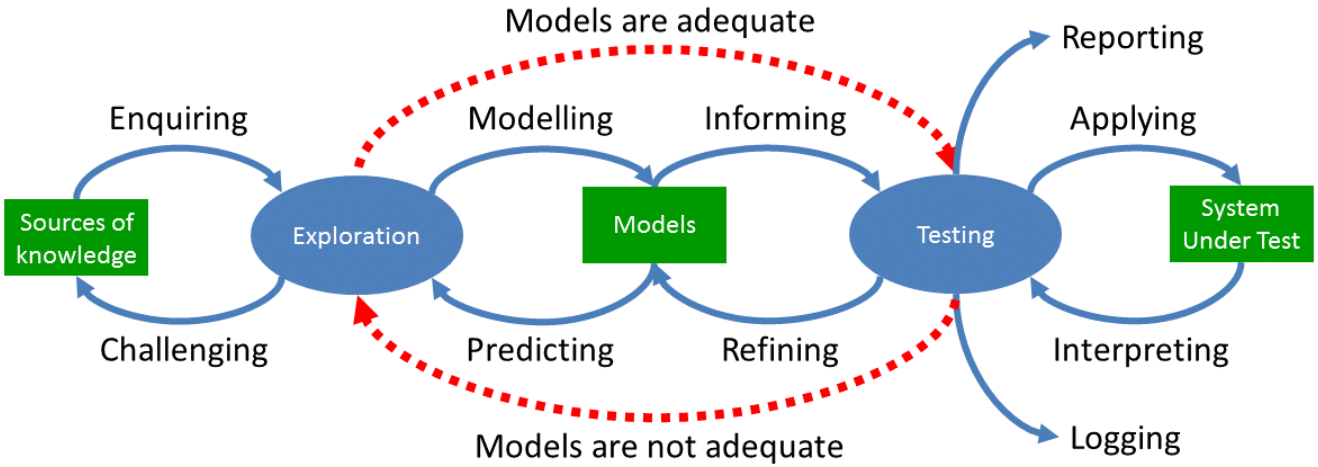
Sources:
People, documents, experience, system under test

Test Models:
Can be documented or mental models

Testing process



New Model Testing

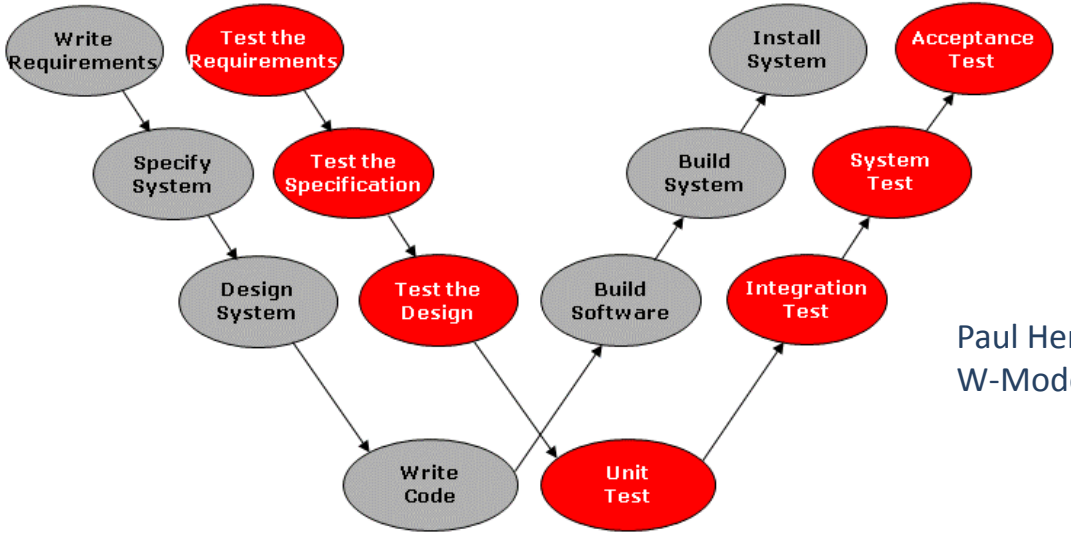


My talk at the BBC: http://blog.gerrardconsulting.com/?q=node/656
29 page paper: http://dev.sp.qa/download/newModel

Shift-left

- Teams redistribute responsibility for testing and collaborate more
- Shift-Left can mean:
  - Developers take ownership for their testing
  - Testers get involved earlier, challenge requirements, share examples with users and devs
  - No test team and no testers
- There is no 'one true way' of course.

Shift-Left is not new



Paul Herzlich's
W-Model 1993

- Shift-Left really brings the thinking about testing earlier in the process
- So, all we do is get involved earlier and ask awkward questions?
- Is it really as simple as that? Well, not quite.

Shift-Left – it's all about feedback

- Testers provide feedback – whenever possible
  - Get involved early – as early as you can
  - Challenge through example
- Software development is knowledge acquisition
  - Knowledge is gathered throughout the project and evolves over time
  - The goal is to assure this knowledge and to ensure it is trusted before it is frozen in code
- Shift-Left is not a threat; it is an opportunity to make a bigger, better contribution.

The Tools Landscape

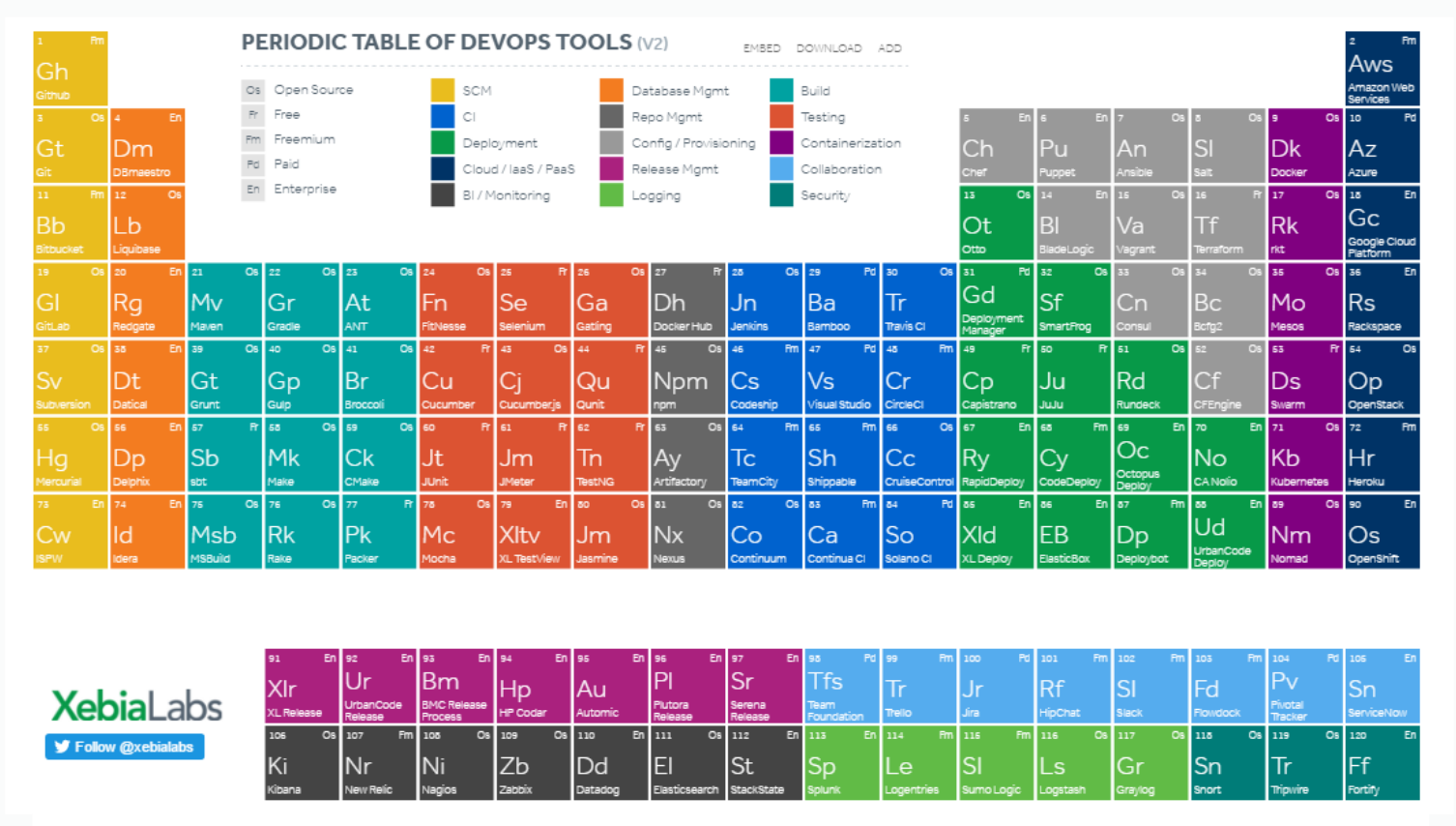
How many tools do you use?

APM - Application monitoring - gives us the eyes on our app and how it's being used / performing
PaperTrail - Log file collector - brings in log files from various servers to one single place - great for systems running across multiple servers
OpsView - Monitoring and alerting tool which we use to bring together monitoring from various systems
Nagios - Used for monitoring and alerting
PagerDuty - Used to alert (SMS and Email and Phone) when a service craps out
Elastic Search, Log Stash and Kibana - Data analysis and monitoring and trending - powerful analyses of what our product is actually doing
Chef - Auto build and deploy technology to allow us to rapidly build and destroy environments (with **Chef Kitchen** and **Knife**)
Vagrant - Create Virtual Environments
Real Time Board - Virtual Whiteboard - amazingly useful
Pivotal Tracker - Agile tracking tool
Fiddler - Proxy web tool
Firebug - Proxy web tool
Zed Attack Proxy - Security testing tool
Burpsuite - Security Testing Tool
HipChat - Real Time IM communication tool
Slack - Real Time IM Communication tool
Rally and **Confluence** - bug tracking and wiki
CloudFormations - Creates templates for Amazon instances

"No doubt we have some more hiding away but that's a pretty good list."

Periodic table of DevOps tools

https://xebialabs.com/periodic-table-of-devops-tools/

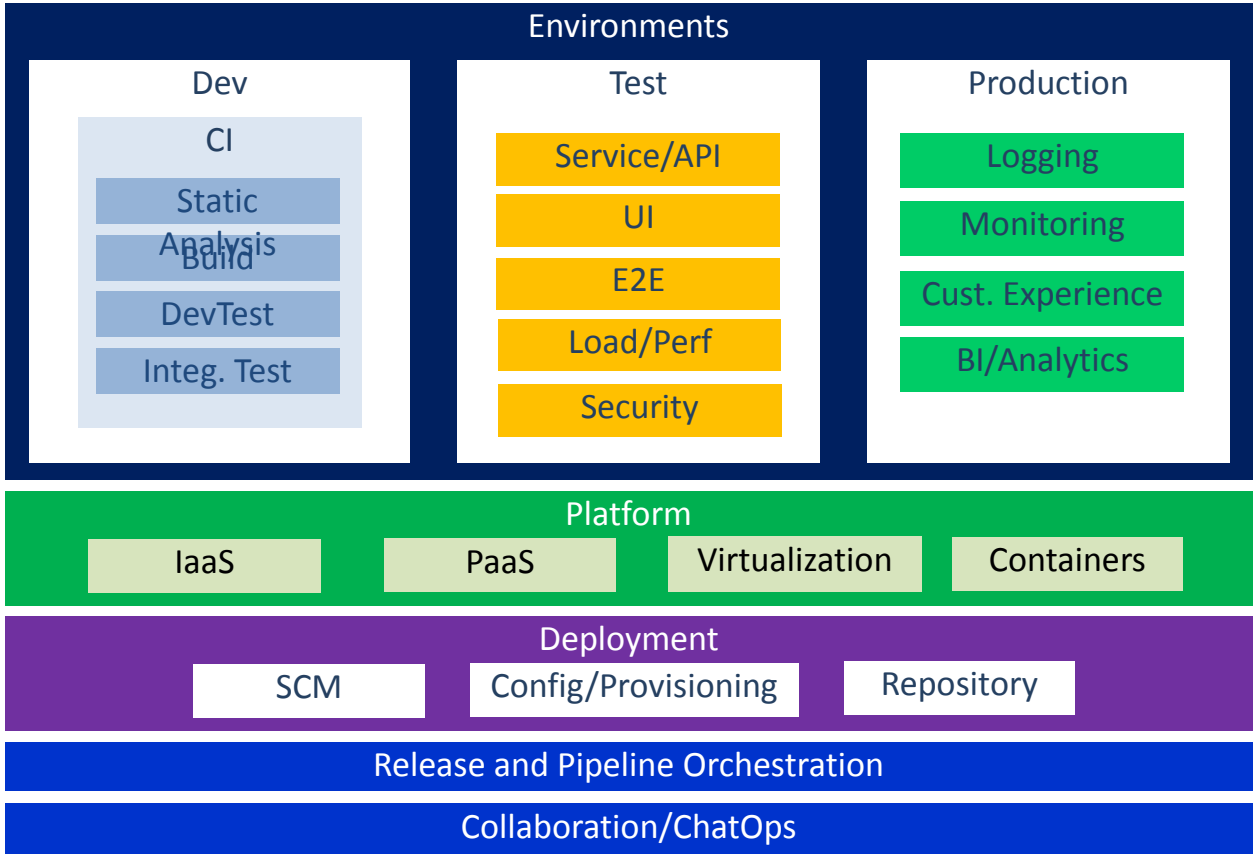


Periodic table of DevOps tools

https://xebialabs.com/periodic-table-of-devops-tools/

Only 100 DevOps tools. We need to include developer, testing and collaboration tools to the landscape

Tools landscape



Tools Knowledge Base (tkbase.com)

- I'm researching tools for tkbase.com
  - 2424 of which 686 are programming languages
  - 1658 tools for DevOps, SDET & Testers
- Tool types and features
  - https://tkbase.com/tools
- My guess is there are at least 2000.

Testing, Analytics and Decision-Making (Shift-Right)

- We test to gather information for someone to make a decision
  - Developers (to fix defects)
  - Project managers (to understand and manage progress)
  - Stakeholders (to be updated and assured)
- In this one respect, testing is all-powerful
- SMAC – Real-Time Analytics
  - Data is analyzed to detect trends, patterns of behaviour, user preferences and opportunities for improvement or new market initiatives
  - Apps instrumented to collect information for decision making.

Modern Practices – Opportunities for Testing

- Shift-Left aims to reduce, if not eliminate, misunderstandings in requirements
- Pervasive automation in DevOps generates much of the data we need automatically
- Results capture and analyses are no longer manual; reporting is almost instant
- Some companies don't log defects or bugs; when defects are found – they are fixed
- But how does testing support decision-making?

Testing and Decision Making

- Testing Uncertainty Principle:
  - *We can predict test status, but not when it will be achieved;*
  - *We can predict when a test will end, but not its status*
- Testing Relativity
  - Stakeholders can't put an absolute value on any test
  - But they can say which test is more valuable
  - So we can use this to scope and proritise
- Quantum Testing
  - Every test adds some quantum of knowledge or it has no value

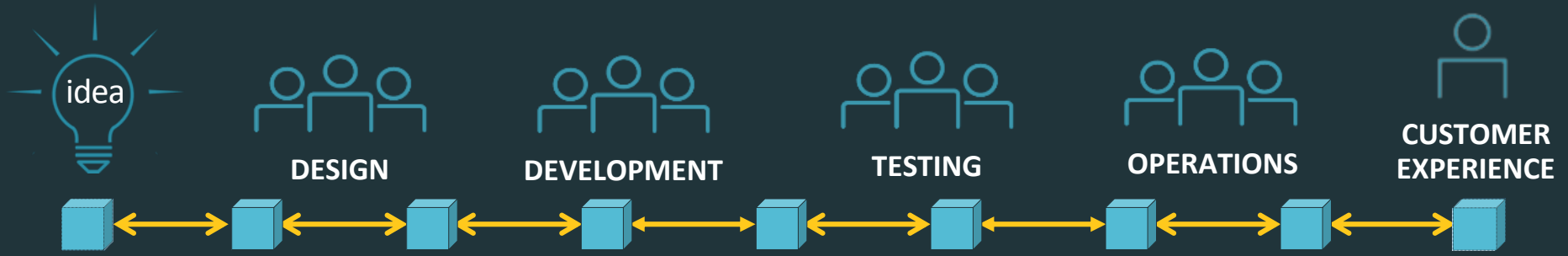
Assurance in the Digital World

- The change that Digital forces on testers and Assurance is profound:
  - We need to re-think how we approach testing so that we achieve levels of confidence in very challenging circumstances.
  - We need ***"power-thinking-tools"*** to create effective tests at volume; it requires both superior modelling skills
  - We need our testers to skill up; to test functionality at scale and with tools almost all the time. The days of manual testing are numbered
- Testing must align with definition and development processes:
  - Shift-Left, to embed and align with developers, to be indispensable partners in the thinking, development and testing processes.

Value | Behaviors | Beliefs, Values, Culture | Heuristics | User Story Mapping | Organization Change | Insight

DesignDevTestOps

Agility across all 4 increases speed, reliability and efficiency



‘Shift-Left’ Digital Evolution feat. DesignOps

‘Shift-Up’ Digital Assurance feat. Cognitive Adaptive Insight

‘Shift-Right’ Digital Ecology

‘Shift-Down’ Digital Archaeology

ca technologies

DesignOps – “Shift Left” (AND Right)

SHIFT LEFT

SHIFT RIGHT

CODE

BUILD

DELIVERY

PRODUCTION

CONTINUOUS ADAPTIVE TESTING

ca technologies

DesignOps – “Shift Left” (AND Right)

SHIFT LEFT

SHIFT RIGHT

CODE

BUILD

DELIVERY

PRODUCTION

CONTINUOUS ADAPTIVE TESTING

ca technologies

Evolution over Revolution

Engineering over Agility

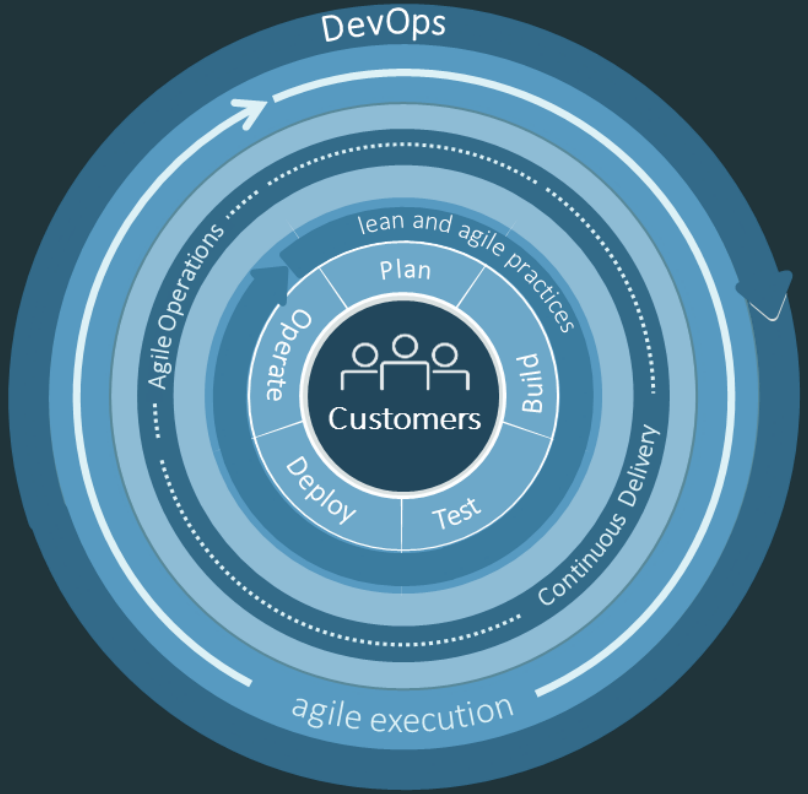
Predictive over Continuous

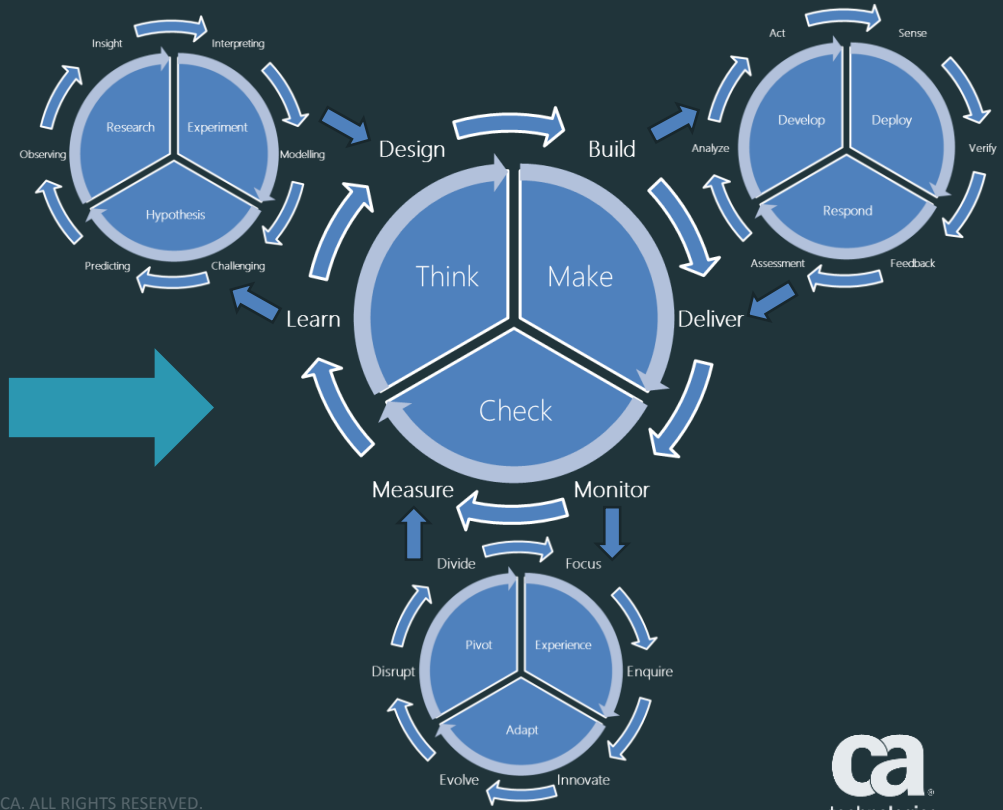
Intelligence over Things

Governance over Ownership

Proactive over Reactive

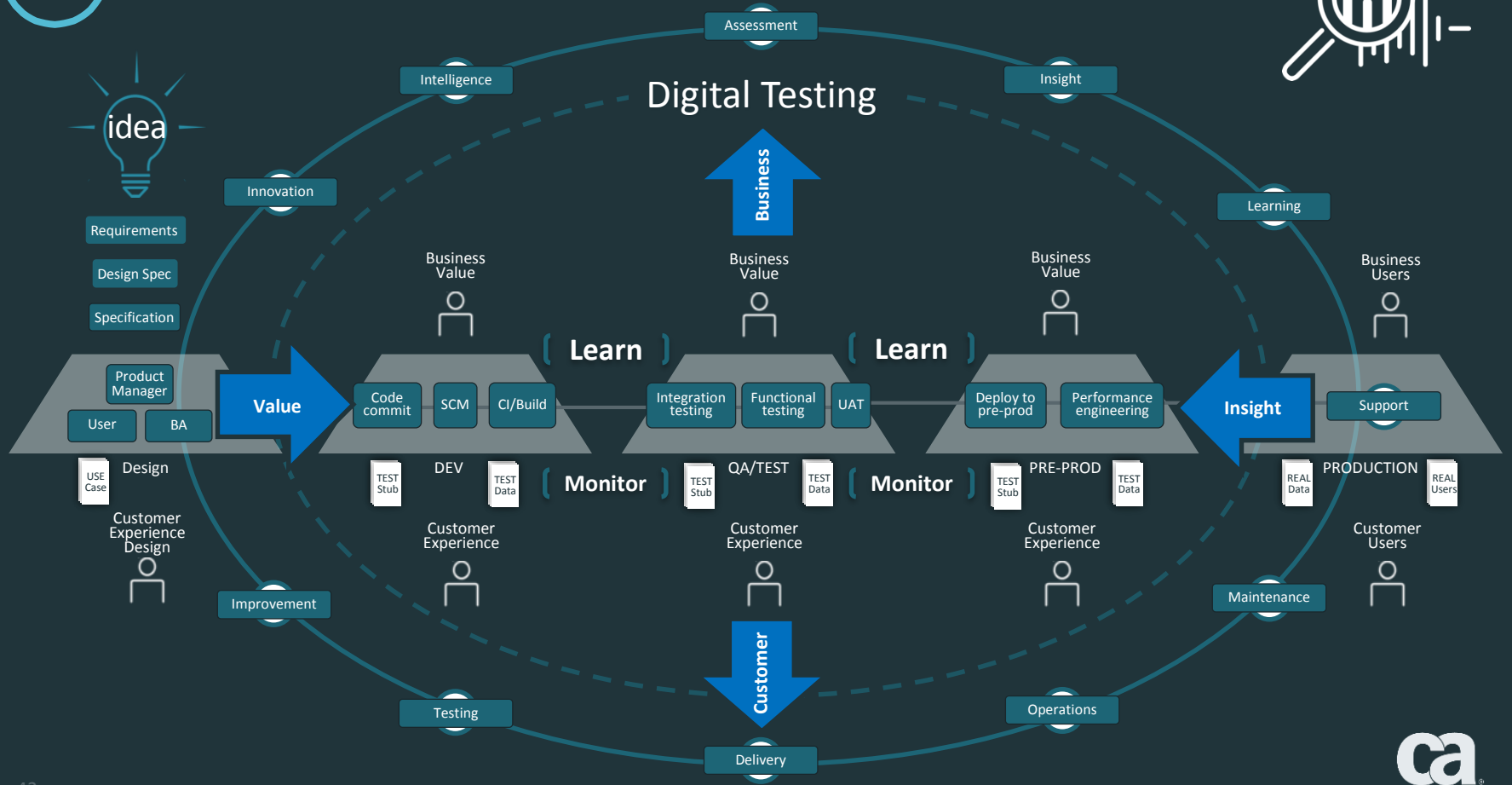
Maturity over Mastery





ca technologies

Digital Assurance



ca technologies

DESIGN → SHIFT-LEFT

DESIGNOPS

SHIFT-RIGHT ← OPS

BPMN
Nimbus
Visio

INTAKE
Requirements
User Stories
Release Plan

TDD
BDD
ATDD
MDD

MODEL
Import User Stories to Automatically Create, Visualize and Optimize Tests

Initiate Functional, Non-Functional and Security as Code

CODE
Develop and Commit Code, Scan Code, Version Control, Continuous Integration. Complete Build and Initiate Release

FUNCTIONAL
- Subset/Mask Test Data
- Create/Reserve Test Data
- Test Automation Library
- Ensure Mobile Experience

NON-FUNCTIONAL
- Performance Engineering
- Simulate Backend Load
- Test Outlier Conditions
- Ensure Mobile Experience
- Security / Penetration

INTEGRATION
Remove Constraints with Virtual Services
- Mobile, Web, App Server, Middleware, Backend, MF
- 3rd Party Systems / API's

Node.Probe ← **Node.Learn**

CONFIG/DEPLOY
- Provision Entire Stack
- Confirm Configurations
- Approve Changes
- Successfully Deploy
- Internal or External Cloud

Node.Test ← **Node.Data**

MEASURE/FEEDBACK
- Customer Experience
- Business Service View
- Application View
- Infrastructure View
- Dynamic Capacity
- Feedback loop

PLAN → AUTOMATION → BUILD → AUTOMATION → TEST → AUTOMATION → DEPLOY → AUTOMATION → RUN

INSIGHT-DRIVEN

Customer Experience

Common Goals
- ✓ Speed/Time-to-Market
- ✓ Quality/Availability
- ✓ Cost/Financial Mix
- ✓ Risk/Compliance

"Big Picture" Issues
- ✓ Requirements
- ✓ Environments
- ✓ Data
- ✓ Automation

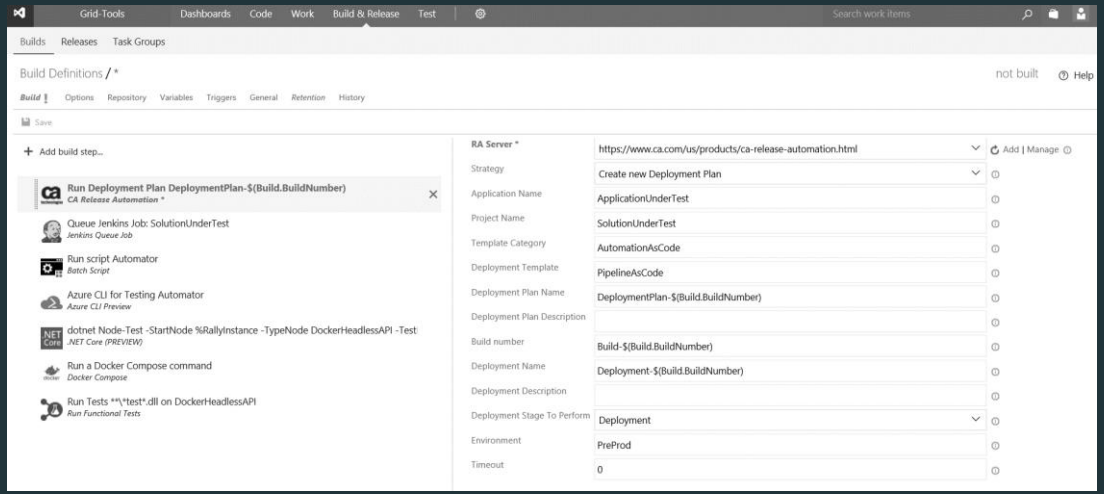
ca technologies

Test in DevOps (TiD)

1. Model-Driven Design of Tests from Use Cases
2. Generate ‘Automation as Code’ from Model
3. Add ‘CA Automator’ Task to ‘Pipeline as Code’

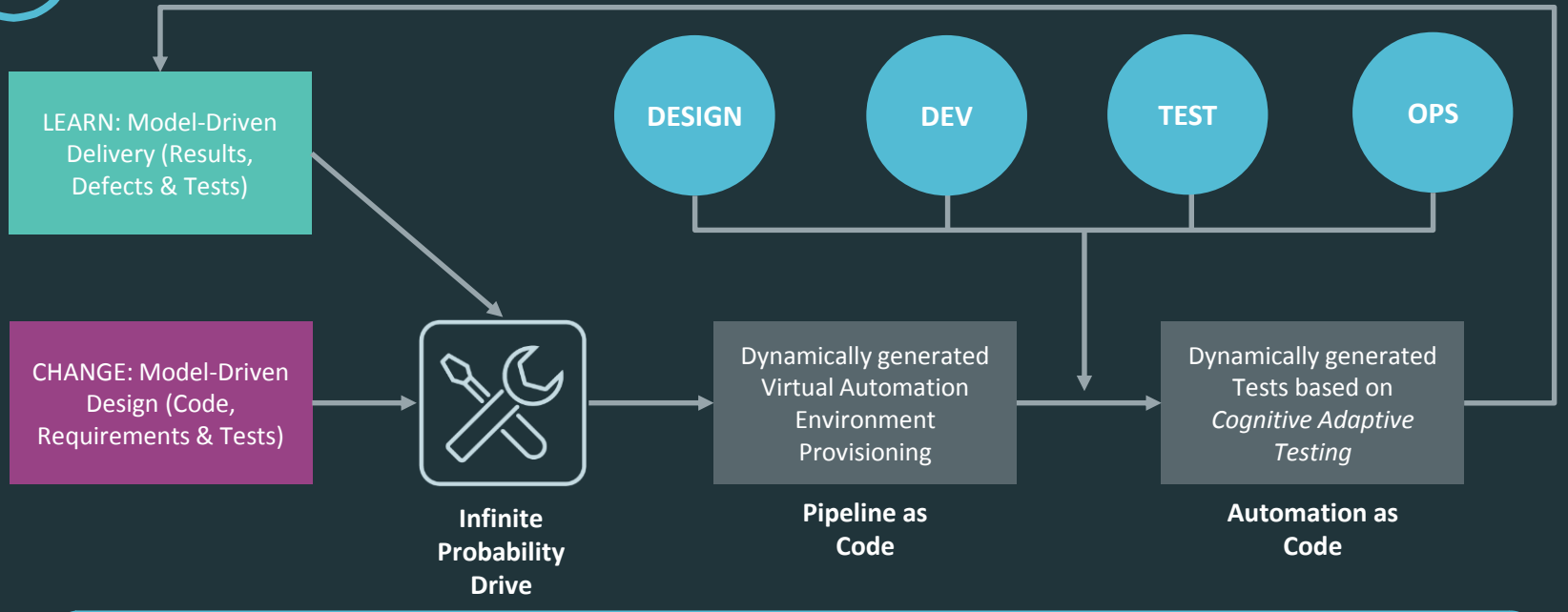
Please can you build tests directly into the pipeline?

“Sure, just assign me the tasks and I will automatically generate the necessary artefacts within whichever release automation platform you want!”




```
PS C:\> Node-Test -StartNode %RallyInstance -TypeNode DockerHeadlessAPI -TestNode %TestSuite -DataNode %TDMInstance -ReleaseNode %Build -EndNode %ArDInstance
```

Infinite Probability Drive




```
PS C:\> Node-Test -StartNode %RallyInstance -TypeNode DockerHeadlessAPI -TestNode %TestSuite -DataNode %TDMInstance -ReleaseNode %Build -EndNode %ArDInstance
```

Pinpoint Failure Analysis


Use Case


Model Flows


Optimize


Generate


Coverage


Workflow


Provision


Code


Iterate


Pinpoint


Analyze


Execute

1. Requirement / Use case / User story (Rally)
2. Generate Model-Driven Design flows (ArD)
3. Optimize & Generate Automation (ArD)
4. Workflow ‘Automation as Code’ (AD Automator)
5. Populate ‘Data as Code’ (ArD Databuilder)
6. Customise ‘Configuration as Code’ (YAML)
7. Build ‘Application as Code’ (Jenkins)
8. Provision ‘Platform as Code’ (Docker)
9. Virtualize ‘Infrastructure as Code’ (SV, NV & NFV)
10. Deliver ‘Pipeline as Code’ (RA CDE, Puppet & Chef)
11. Deploy Headless Test Runners (MicroContainers)
12. Deprovision ‘Environment as Code’ (GIT / Blob)
13. Analyse ‘Results as Code’ (ArD Server)
14. Pinpoint Failure (Optimizer NLP)
15. Generate new Model-Driven Delivery flows (CLI)

‘Pipeline as Code’

Automation

Infrastructure

Platform

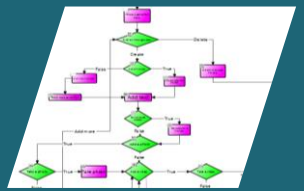
Microservices

Release

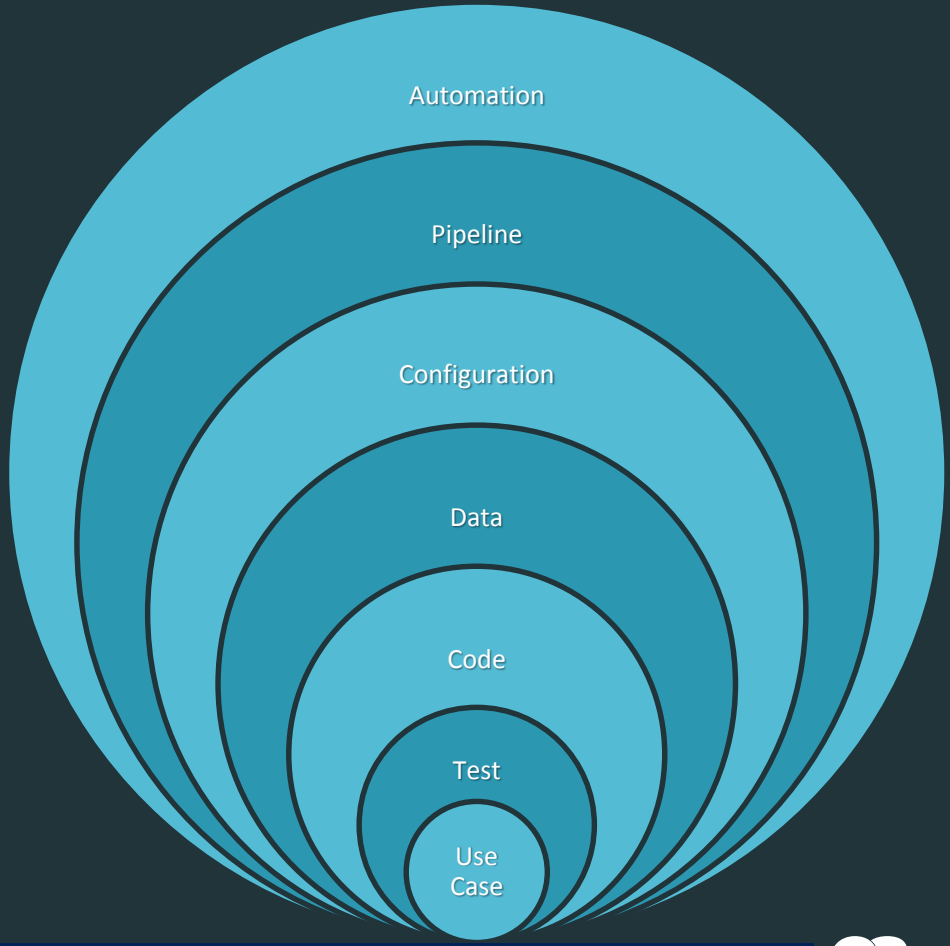
Environment

Applications

Model



Command Line Interface for Testers




```
PS C:\> Node-Test -StartNode %RallyInstance -TypeNode DockerHeadlessAPI -TestNode %TestSuite -DataNode %TDMInstance -ReleaseNode %Build -EndNode %ArDInstance
```

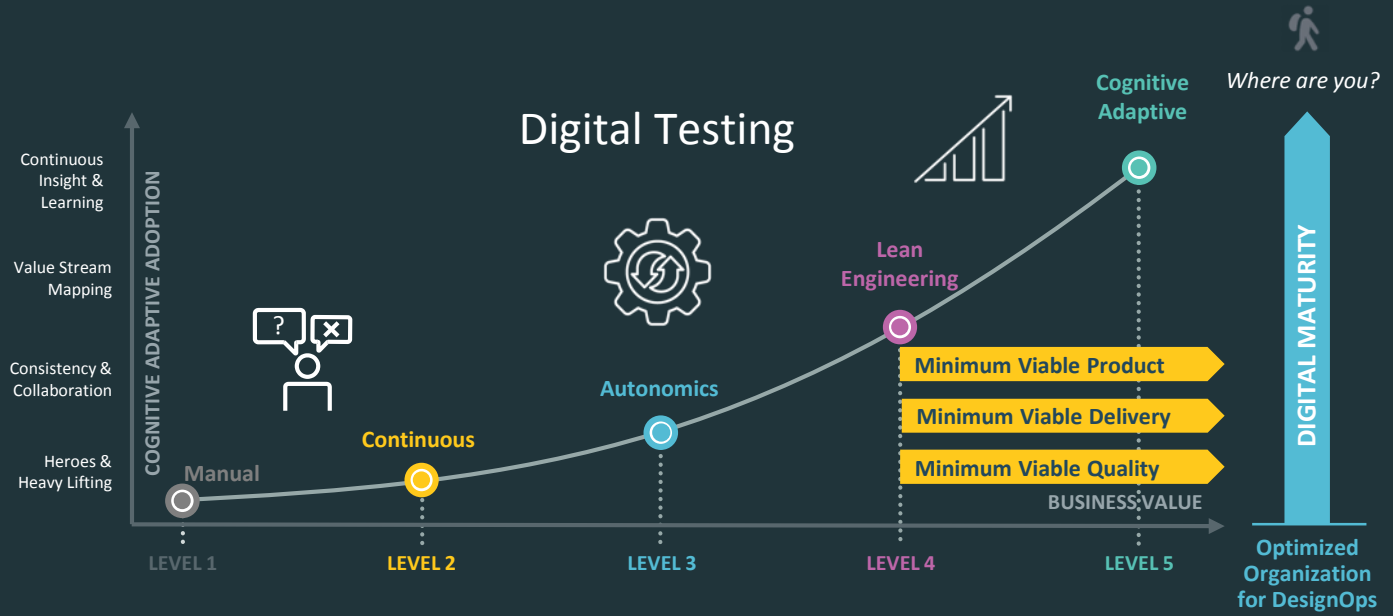

ca technologies

‘Automation as Code’

*CA Agile Requirement Designer - Automation Builder has native support for EggPlant, Ranorex, Selenium 3.0 and Cucumber.

```
PS C:\> Node-Test -StartNode %RallyInstance -TypeNode DockerHeadlessAPI -TestNode %TestSuite -DataNode %TDMInstance -ReleaseNode %Build -EndNode %ArDInstance
```

Adoption | Learning | Intelligence | Innovation | Improvement | Testing | Delivery



- Silos, manual handovers, waterfall process
- One release/year
- Monolithic apps
- Long term project/resource planning
- Error prone dev/test/release processes

- Insight-Driven (Predictive / Prescriptive)
- NoOps organized (IoT-Ops/DX-Ops/Sec-Ops)
- Self-healing end-to-end autonomic orchestration
- Microcontainerization & Microservices enabled
- Open Innovation & Connected Intelligence
- Quantum Teleportation (Shift X)

The shift towards Value-Driven Delivery

Cognitive Adaptive Adoption (CA-A)

Manage & Monitor

Make a great customer experience a competitive advantage

Minimum Viable Experience

CA Project & Portfolio Management

Unify long term strategy, investment and portfolio planning.

Complexity-Informed Organizational Change

Release & Deploy

Control the release process, to continuously advance application quality, improve the customer experience and reduce costs.

Continuous Delivery

Minimum Viable Product

CA Agile Central

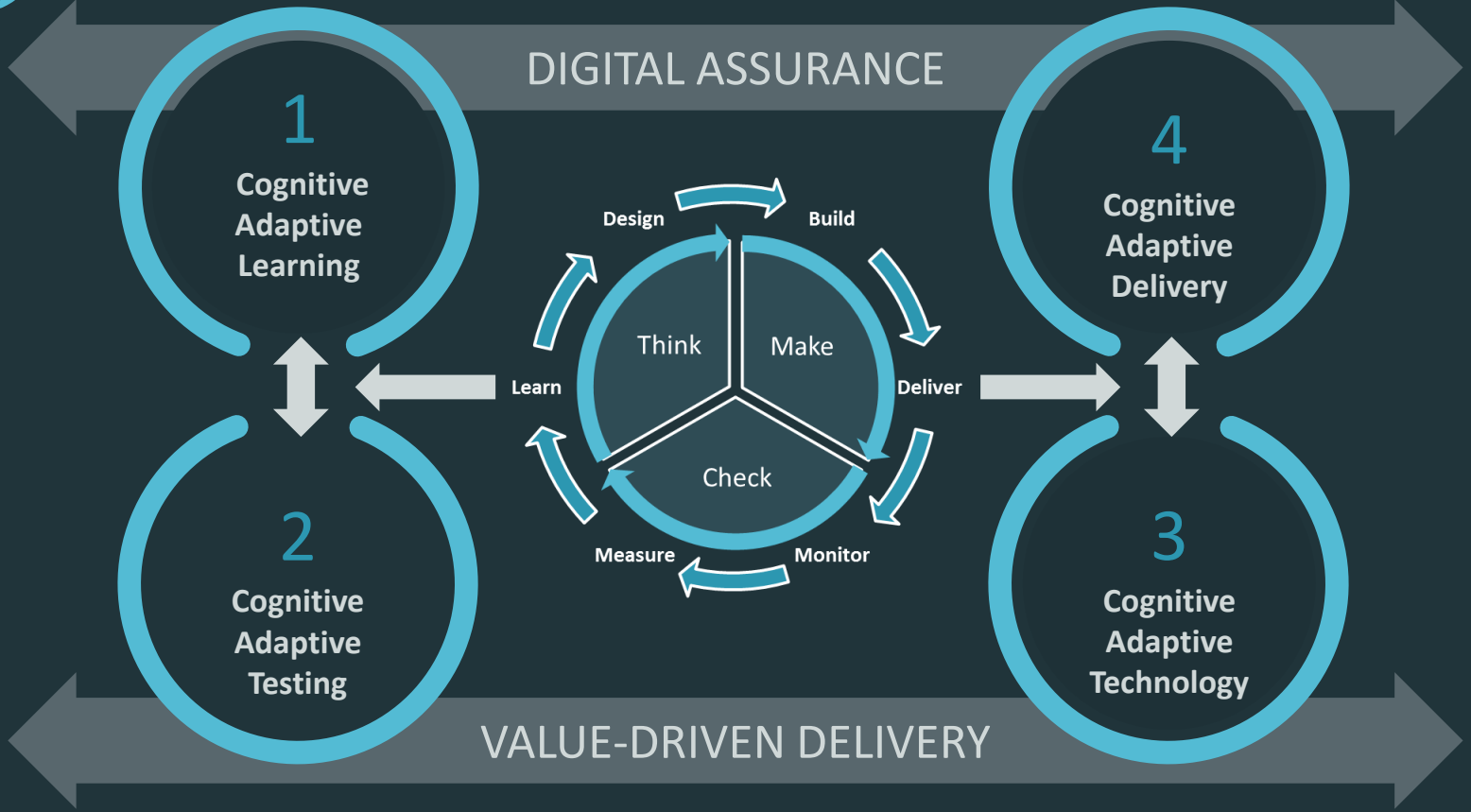
Collaboratively plan, prioritize and track work across the enterprise.

Develop & Test

Synchronize efforts to dramatically speed app development & increase quality

Minimum Viable Quality

Cognitive Adaptive Adoption (CA-A)



ca technologies

Cognitive Adaptive Insight (iCA)

Powered by Actionable Intelligence technologies

REAL TIME, BIG DATA ANALYTICS EMBEDDED THROUGHOUT OUR PORTFOLIO

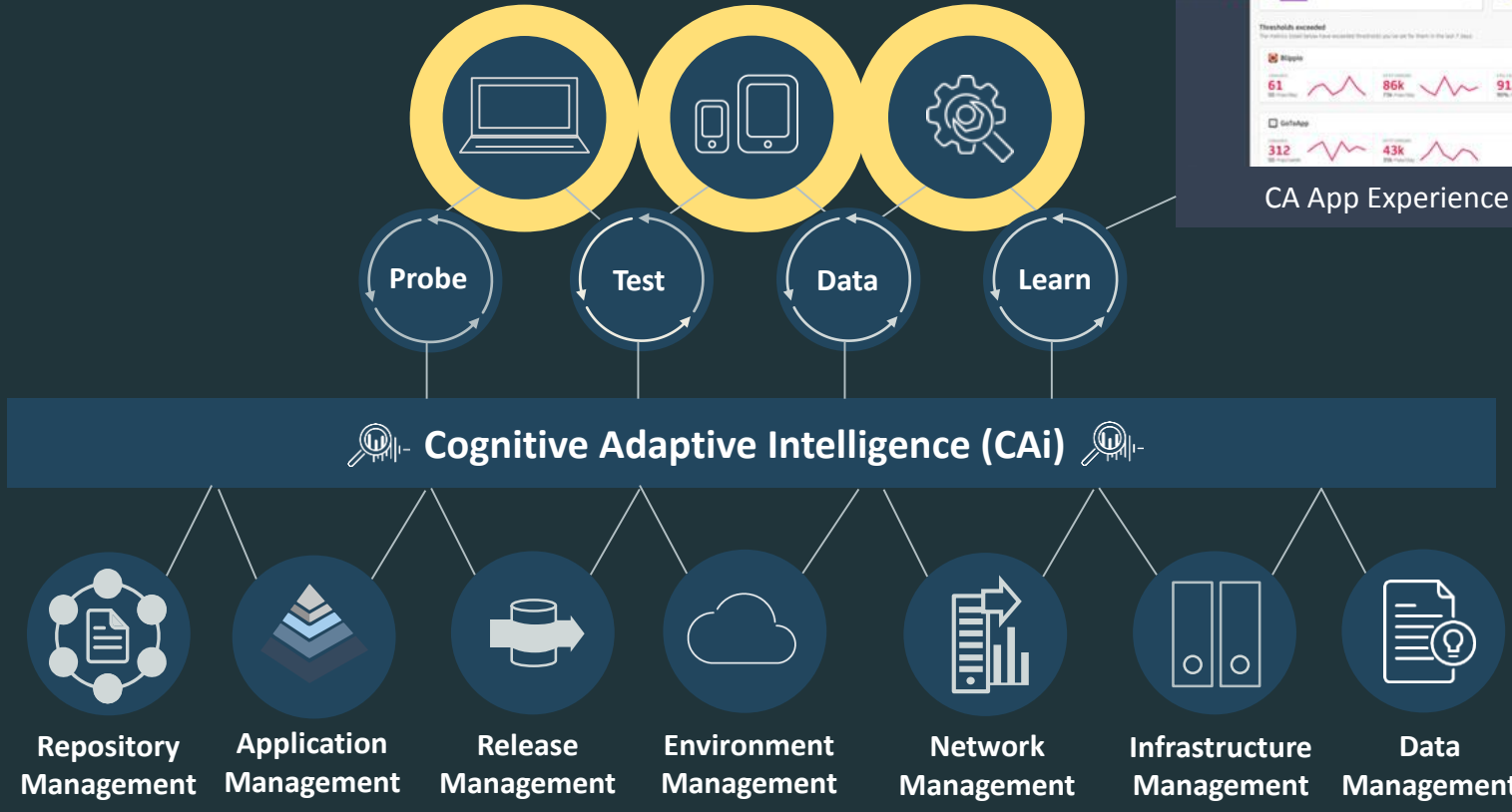
AGILE MANAGEMENT	AGILE OPERATIONS	DEVELOPER PRODUCTS
CONTINUOUS DELIVERY	MAINFRAME	SECURITY
Release management dashboard	Real-time app performance	Access incident response	API analytics

ca technologies

Cognitive Adaptive Insight (iCA)

Powered by Actionable Intelligence technologies

CA App Experience Analytics



ca technologies

Cognitive Adaptive Intelligence (CA-I)

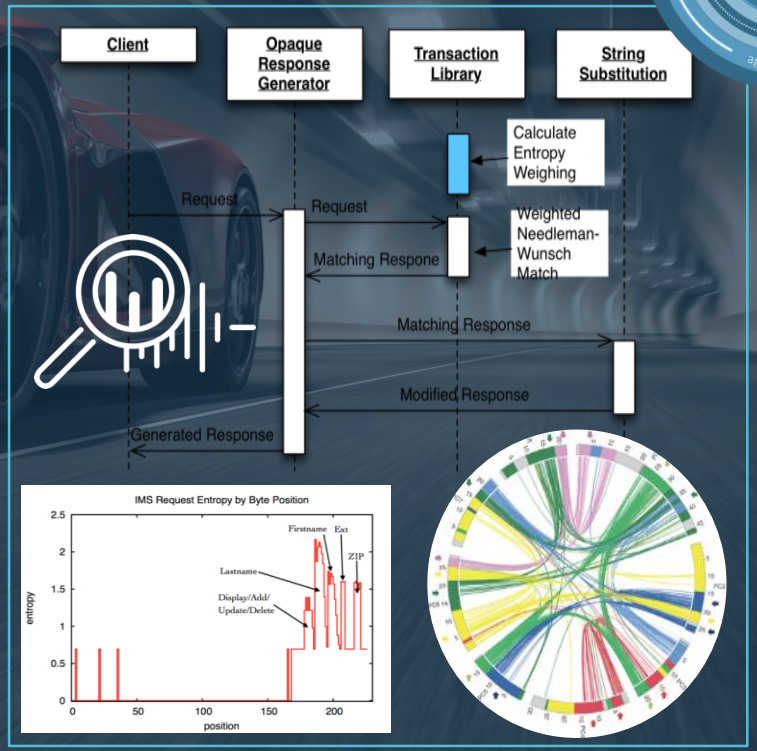
Use Case – CA Service Virtualization

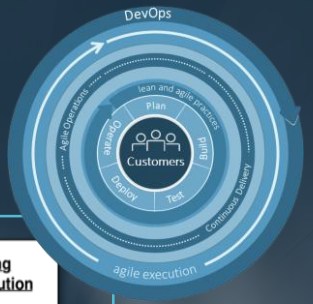
BRINGS TRUE ARTIFICIAL INTELLIGENCE (AI) TO LIFECYCLE VIRTUALIZATION

- Virtualize services without requiring any knowledge or decoding of the service protocols
- Applies a genome sequence alignment algorithm, discovers byte-level patterns in message protocols
- Now virtualize a much wider range of protocols without requiring a new DPH

HIGH ACCURACY: 99.6 – 100%

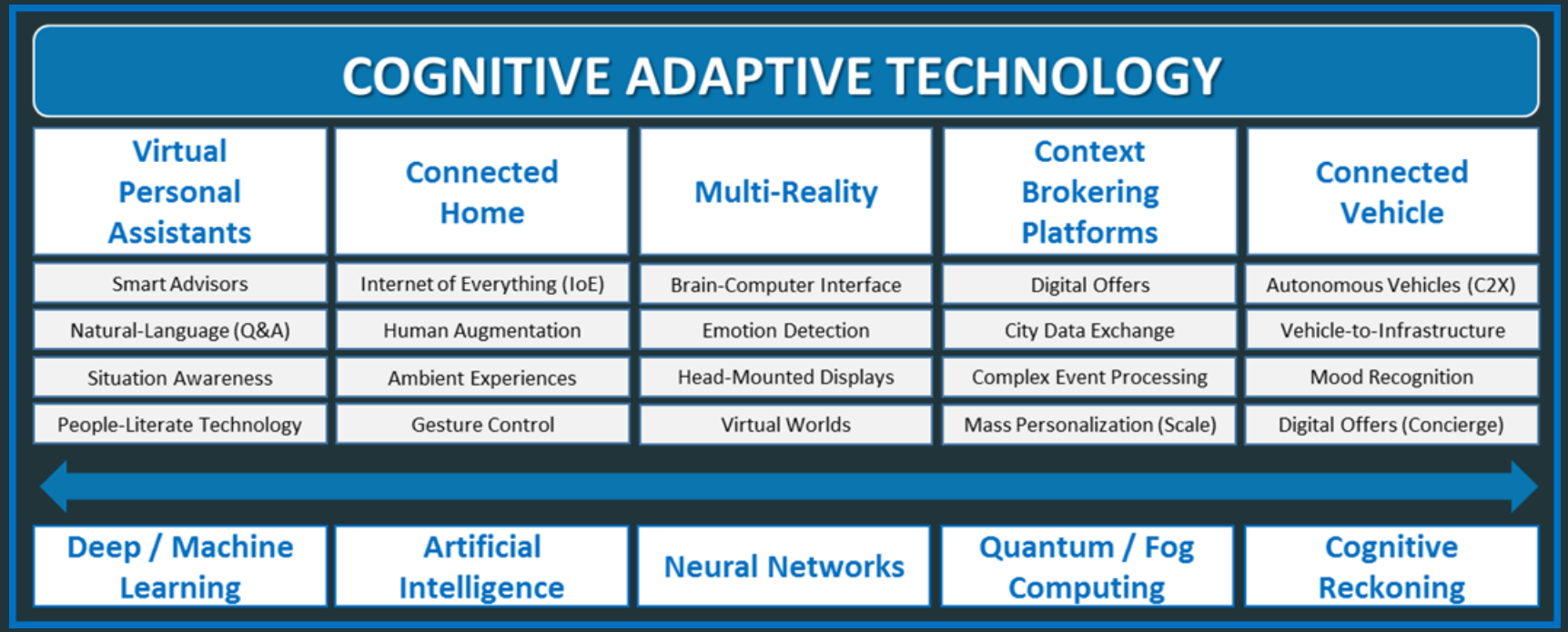
- Increased speed and accuracy with Entropy Weighting + Message Clustering
- The more data a service observes the data, more intelligent it becomes
- Perfect for performance testing where we deal with tons



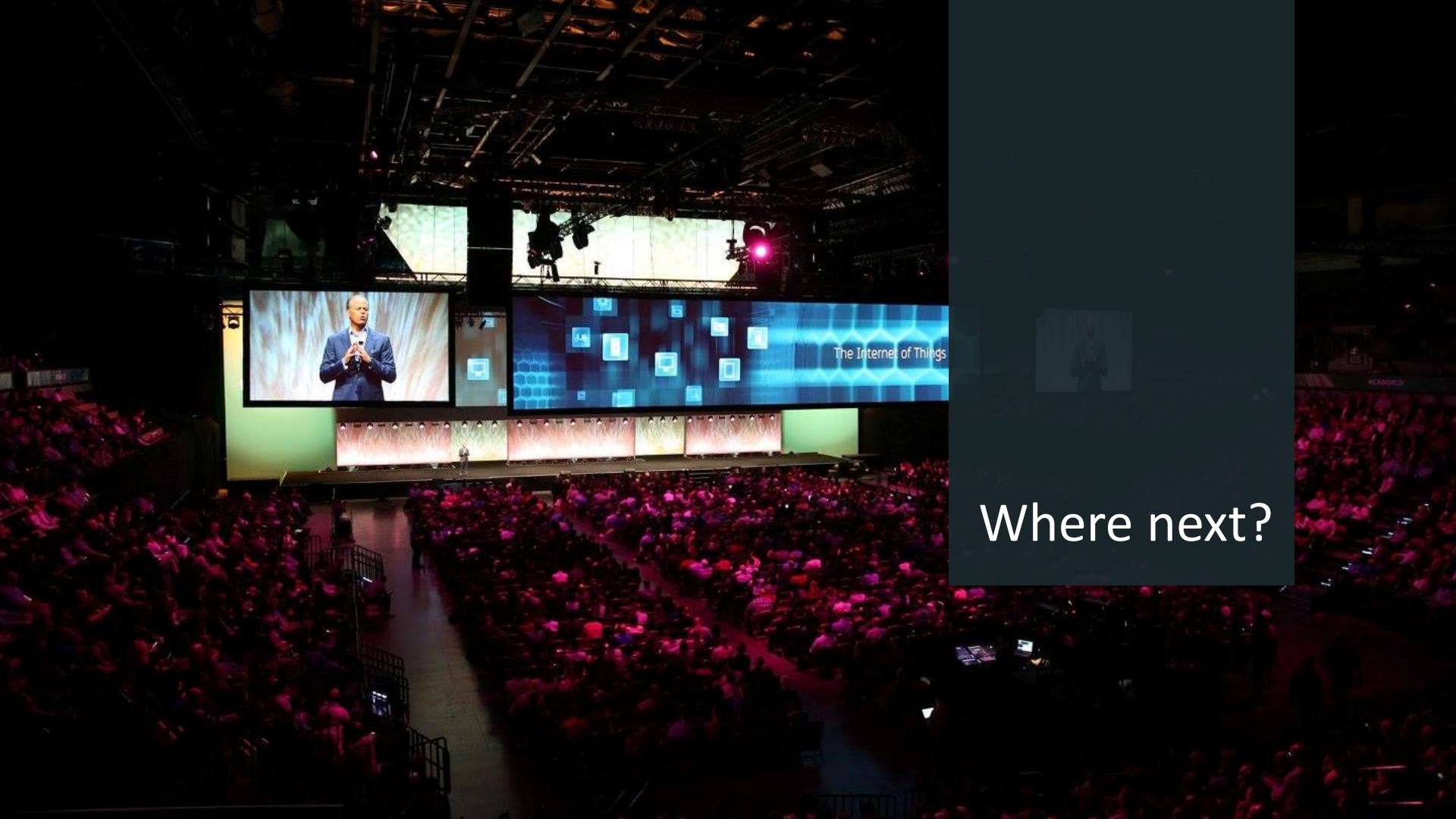


Cognitive Adaptive Technology

The next generation of Knowledge, Vision, Adaptive, Speech & Search (KVASS)



1. Digital Mobile, PerfectoMobile, Chapter 22, Jonathon Wright



Where next?

Where next?

Continuous Adaptive Testing

CA Technologies

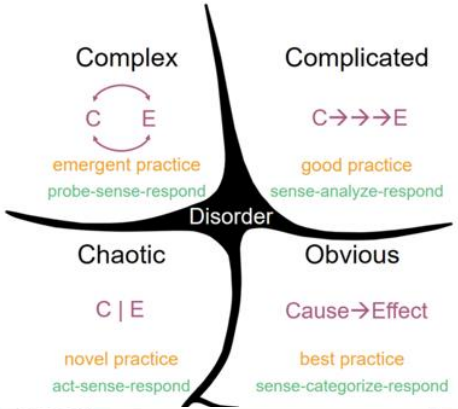
Blogs - #ExcuseFreeTesting

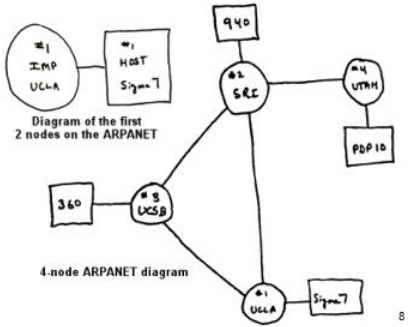
Webcast Series #1 - Continuous Testing 101

Webcast Series #2 - Continuous Performance 101

Blogs #ExcuseFreeTesting – Legacy is your Legacy!

Last month when live on stage at STARWest in California, I was able to ask Alexa to 'turn on my heating' back in the UK.





This abstraction layer will enable data scientists, statisticians and engineers to enable value and insight-driven predictive, prescriptive and deep learning, it achieves this by a process of machine learning, training algorithms

https://dzone.com/articles/legacy-is-our-legacy-1

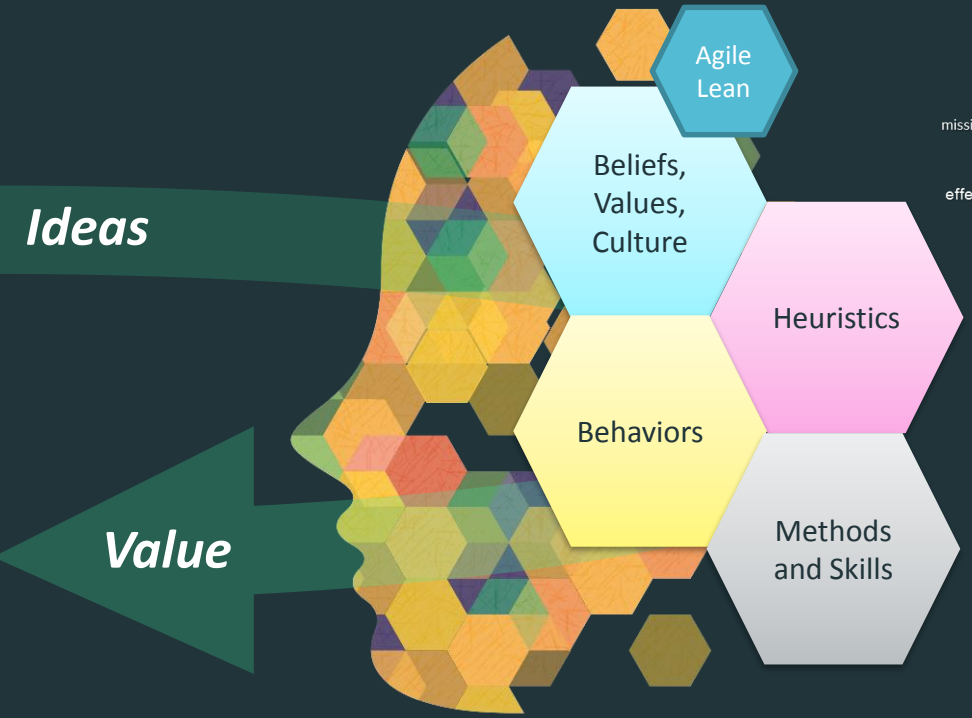
:y Network (ARPA) Network, 1969, 2 & 4 Nodes (the first the Internet

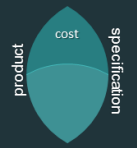
[2] STARWest, 'Think you can just test that API? think again!', 6th October 2
http://www.slideshare.net/Jonathon_Wright/starwest-think-you-can-jus
[3] 'The Turing Bombe (Enigma)', Bletchley Park, Todd DeCapra, Author of Engineering, http://www.lysator.liu.se/~koma/turingbombe/
[4] 'Calculating Ada: Countess of Computing', BBC4, Macclesfield, Hannah

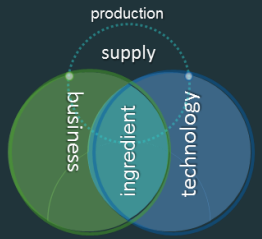
Solution Thinking

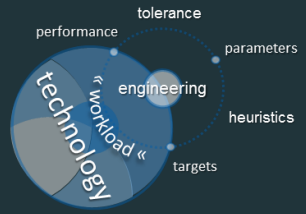
Webcast Series - Continuous Testing 101

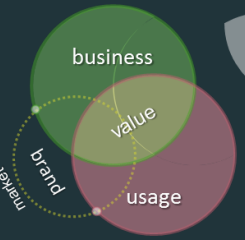
Ray Arell

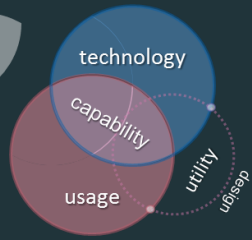










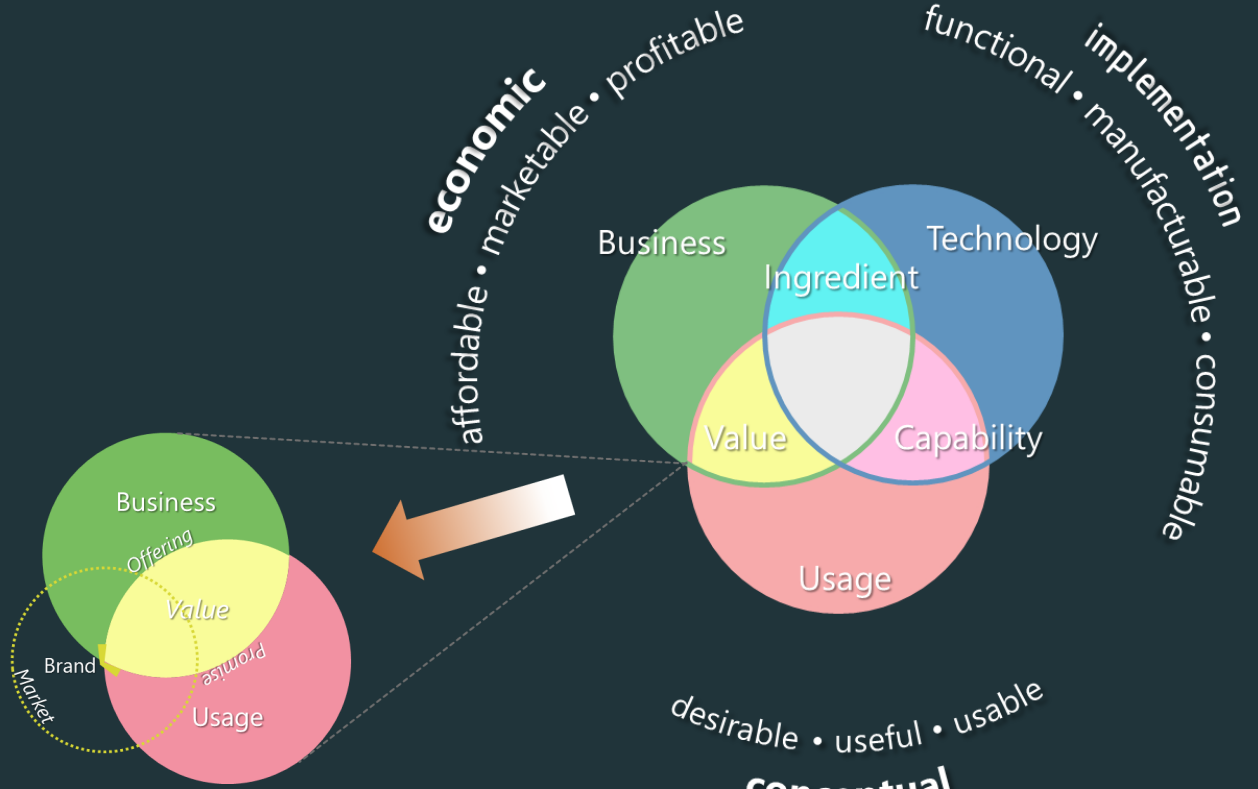


ca technologies

Value-Driven Delivery

Webcast Series - Continuous Testing 101

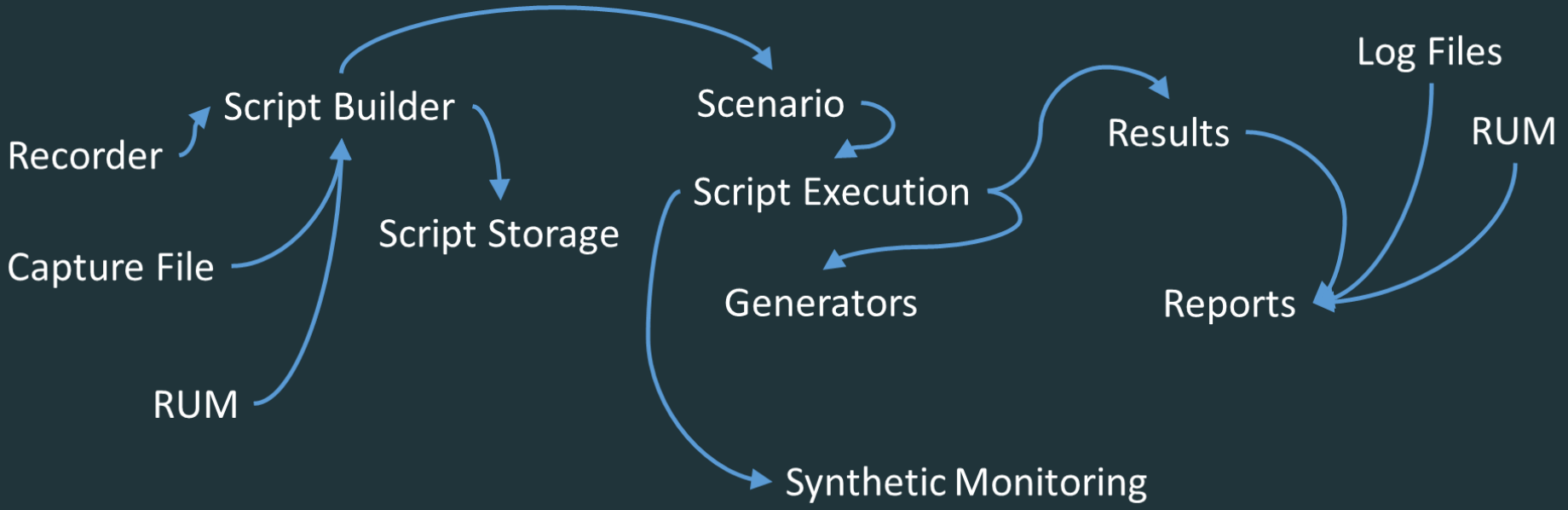
Erik Simmons



Performance Engineering

Webcast Series - Continuous Performance 101

Wilson Marr



ca technologies

Q & A

ca technologies