



The three-pillar, balanced quality model of...

...Proactive QA, Detective QA and
Reactive QA

Szilard Szell - *DevOps Transformation Lead* - 27.11.2024

www.eficode.com



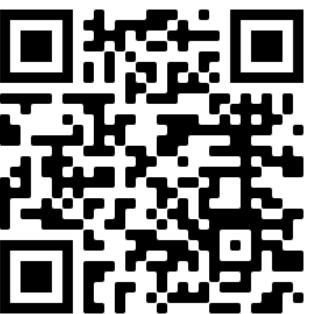
Szilárd Széll

- ✓ DevOps transformation lead
- ✓ Test and Quality Coach
- ✓ Agile coach and SAFe SPC, trainer
- ✓ Volunteer in ISTQB
- ✓ Public speaker

Experience

- 24 years of experience in QA and DevOps in Telecommunications industry
- 12 years of experience as change agent
- SAFe SPC, Certified Scrum Master, DevOps DASA
- ISTQB CTEL-ITP-Full, CTAL-TM, CTFL-AT, CTFL, IREB CPRE
- ITIL4 Foundation
- Lean Six Sigma Green Belt
- Lean Service Creation - Facilitator
- XRAY Certified Expert

“Testing is learning about your product and giving feedback. Continuous Testing is amplifying feedback”



eficode

The three ways of DevOps and Quality



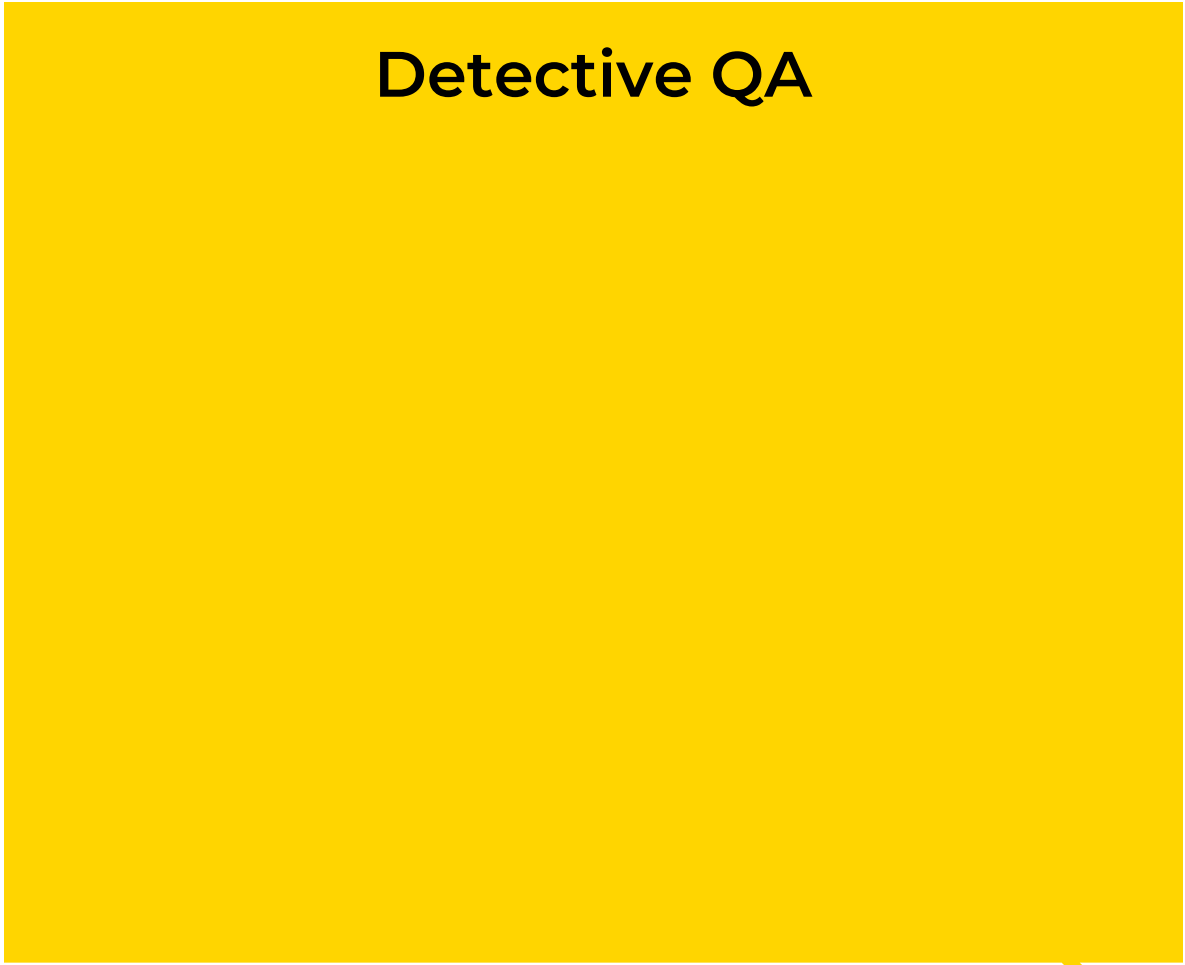
1. The First Way: The Principles of Flow



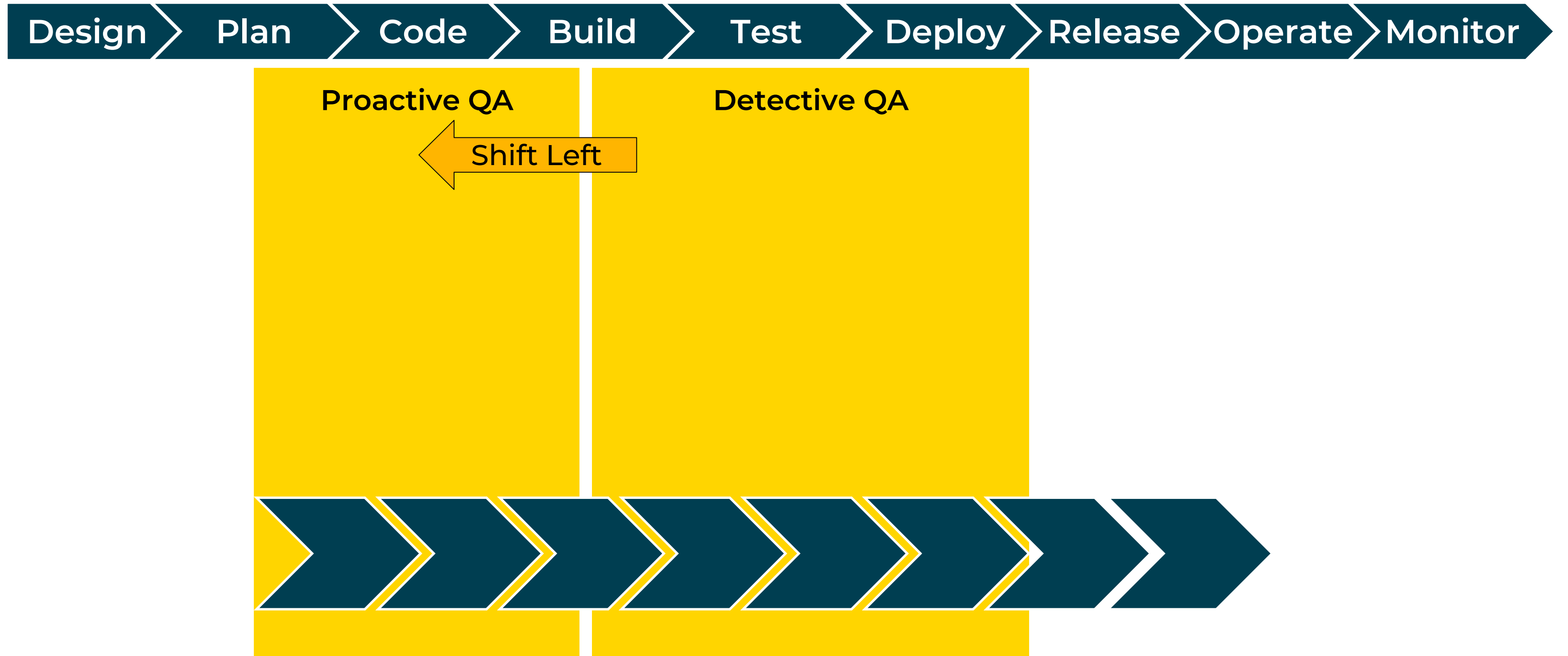
“Being able to take needless work out of the system is more important than being able to put more work into the system.”

Gene Kim

More testing slows down the flow



Automation and Shift Left for faster flow

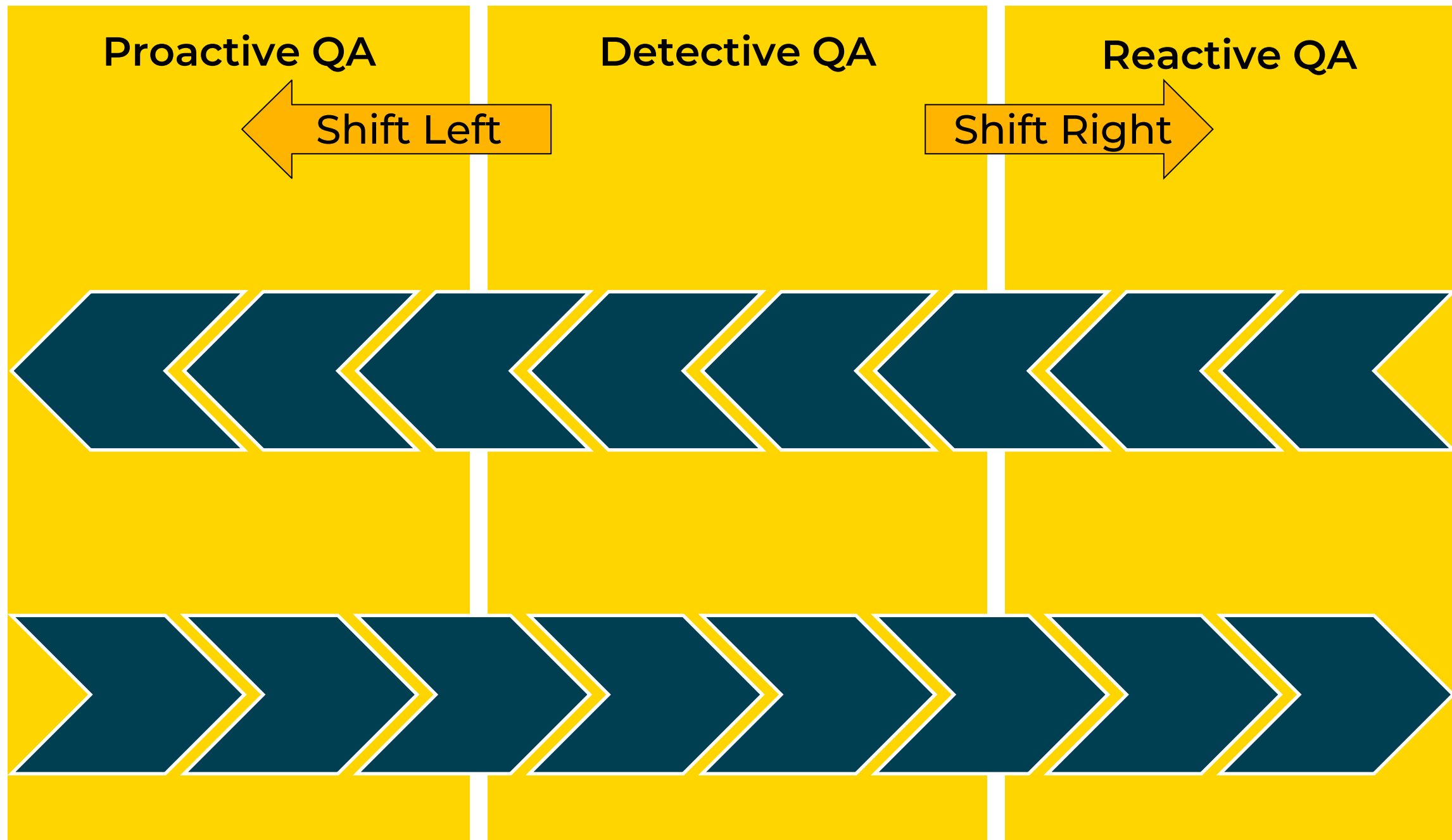


2. The Second Way: The Principles of Feedback

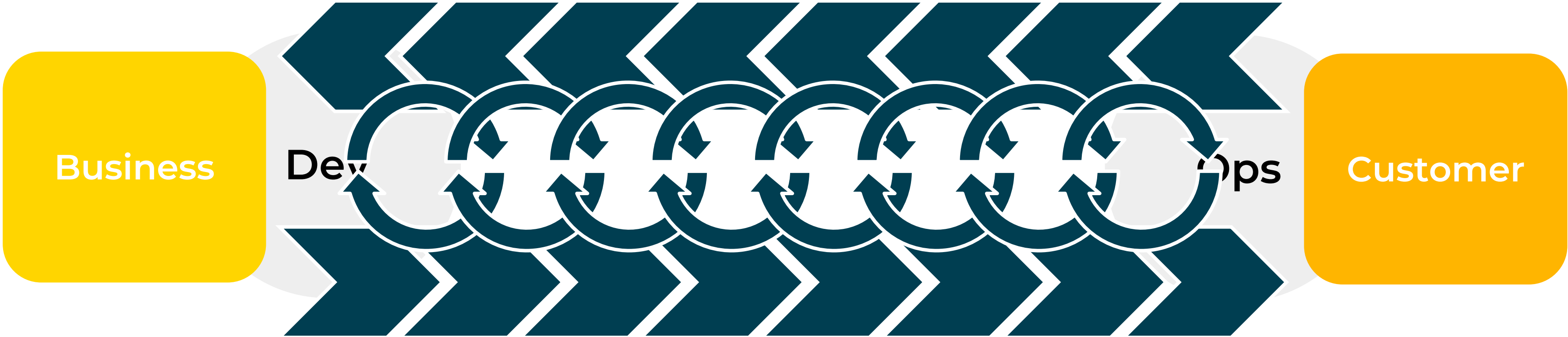


“Improving daily work is even more important than doing daily work.”

Shift Right to amplify Feedback



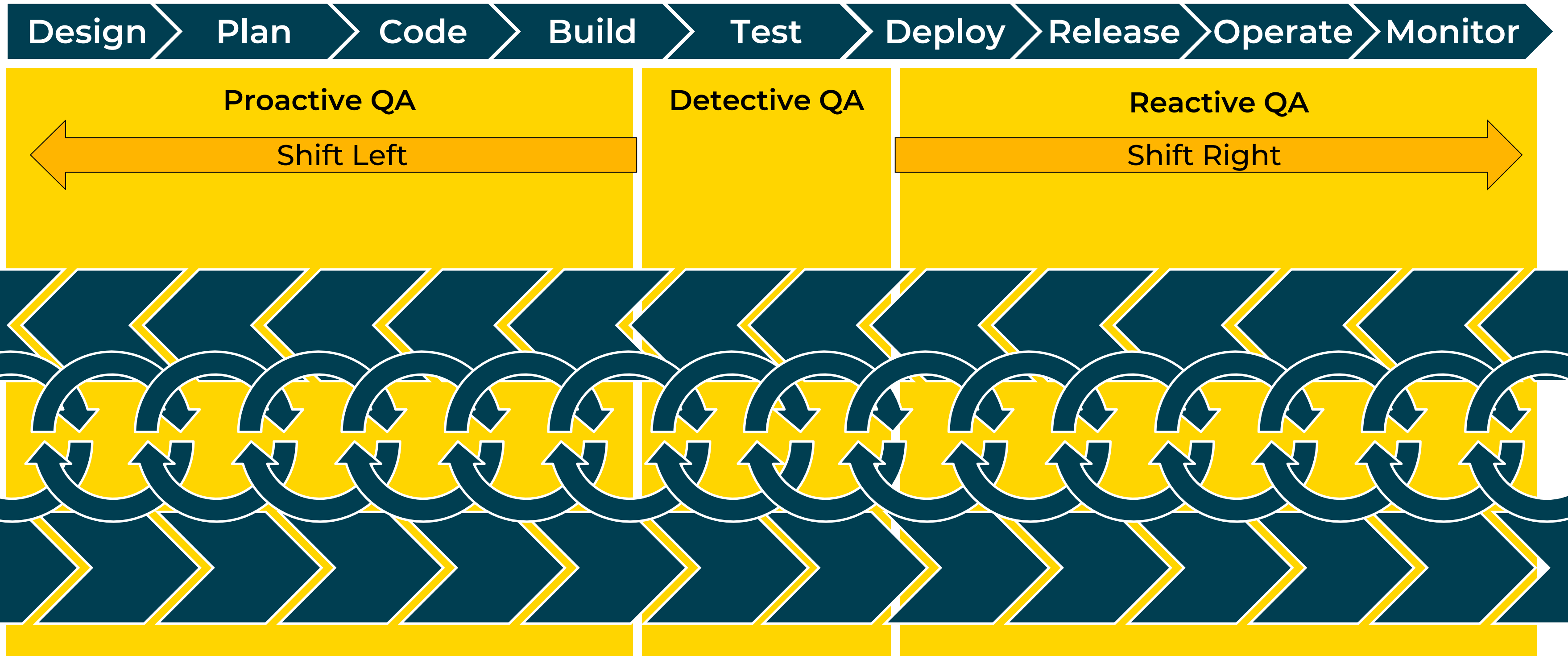
3. The Third Way: The Principles of Continual Learning and Experimentation



“If you can’t out-experiment and beat your competitors in time to market and agility, you are sunk.”

Gene Kim

All to the Left and Right to amplify Learning





Quality Transformation



Quality transformation



Where you want to be

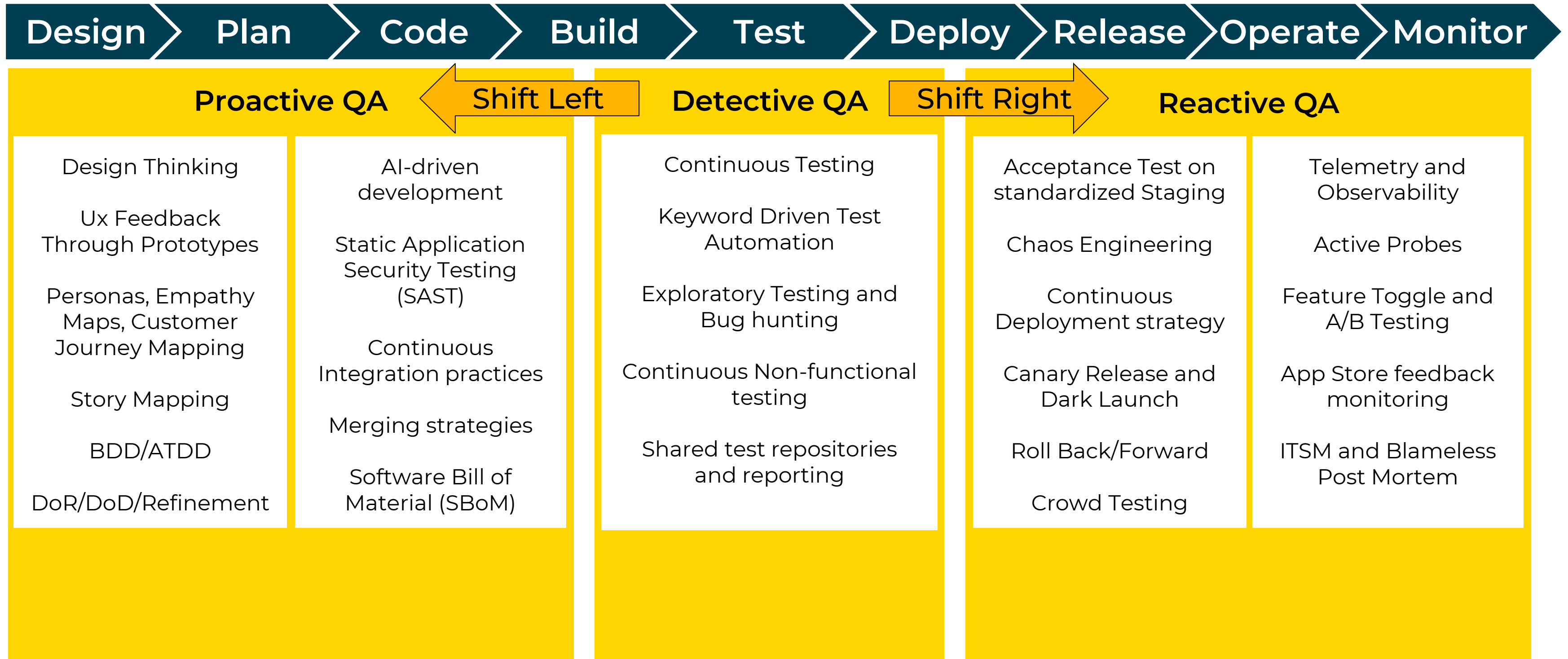
Where most companies focus

Where you want to be



When designing:	When coding:	When testing:	When releasing:	When operating:
Understanding user needs well. Writing development requirements based as tests.	Using AI-driven development to avoid human error. Continuous Integration practices to release and test small increments.	Using automated tests and best practices like shared test repositories and reporting standards.	Getting acceptance testing and early user feedback through pilots and crowd testing Prepared for plan A/B/C	Monitoring the data, getting feedback, and finding improvement possibilities.

Adding all practices. Are we there yet?



Agile Practices = Built in Quality

Automation = Fast Feedback

Transparency = Trust

Internal Developer Platform

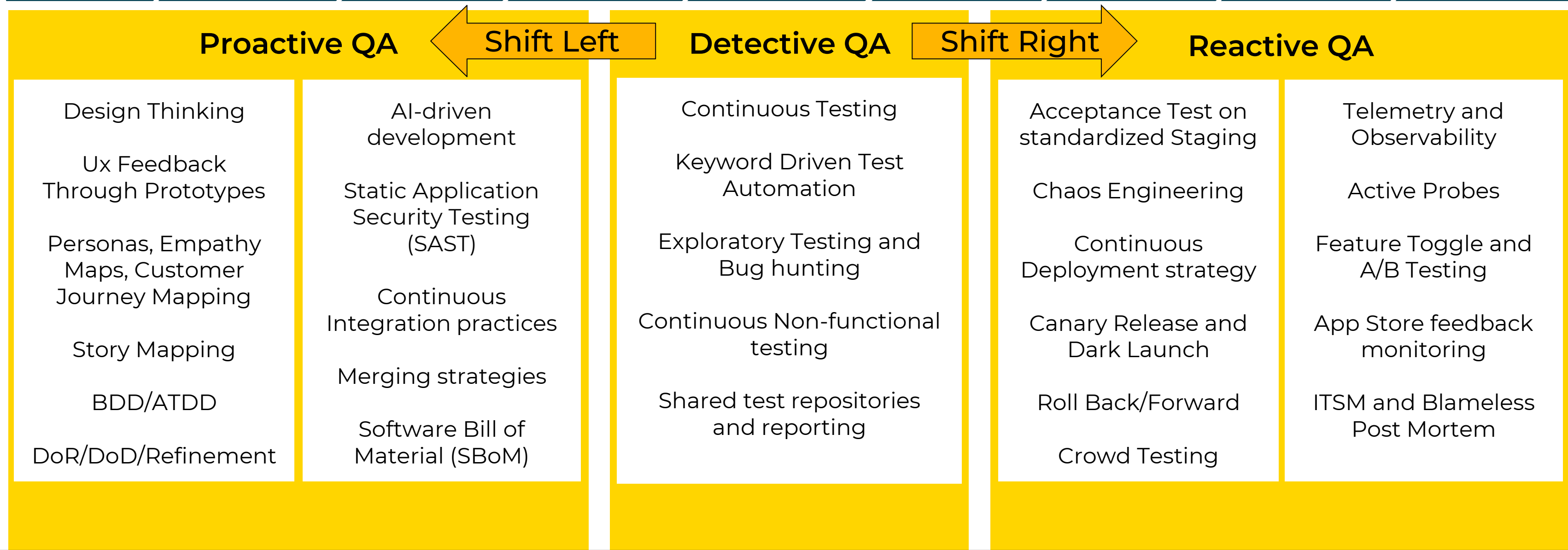


A foundation of self-service APIs, tools, services, knowledge and support which are arranged as a compelling internal product

Evan Bottcher

Head of Engineering at Thoughtworks

Platform engineering for efficiency



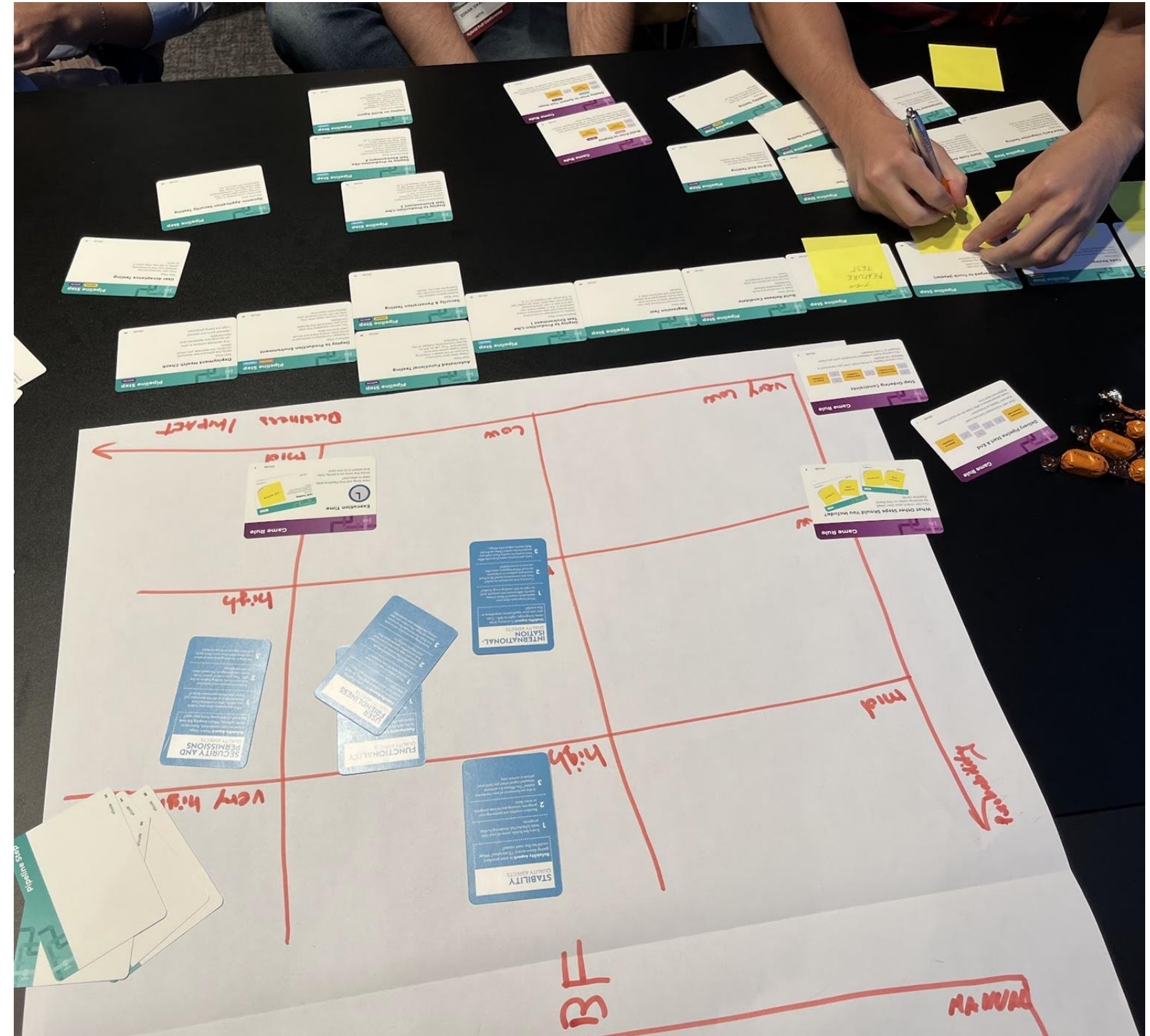
Internal Developer Platform

Agile Practices = Built in Quality

Automation = Fast Feedback

Transparency = Trust

Collaborate on CQA Strategy as pipeline



Continuous QA Strategy as pipeline



Proactive QA		Shift Left	Detective QA	Shift Right	Reactive QA
Design Thinking	AI-driven development		Continuous Testing	Acceptance Test on standardized Staging	Telemetry and Observability
Ux Feedback Through Prototypes	Static Application Security Testing (SAST)		Keyword Driven Test Automation	Chaos Engineering	Active Probes
Personas, Empathy Maps, Customer Journey Mapping	Continuous Integration practices		Exploratory Testing and Bug hunting	Continuous Deployment strategy	Feature Toggle and A/B Testing
Story Mapping	Merging strategies		Continuous Non-functional testing	Canary Release and Dark Launch	App Store feedback monitoring
BDD/ATDD	Software Bill of Material (SBoM)		Shared test repositories and reporting	Roll Back/Forward	ITSM and Blameless Post Mortem
DoR/DoD/Refinement				Crowd Testing	

CQA Strategy implemented in the CI/CD pipeline

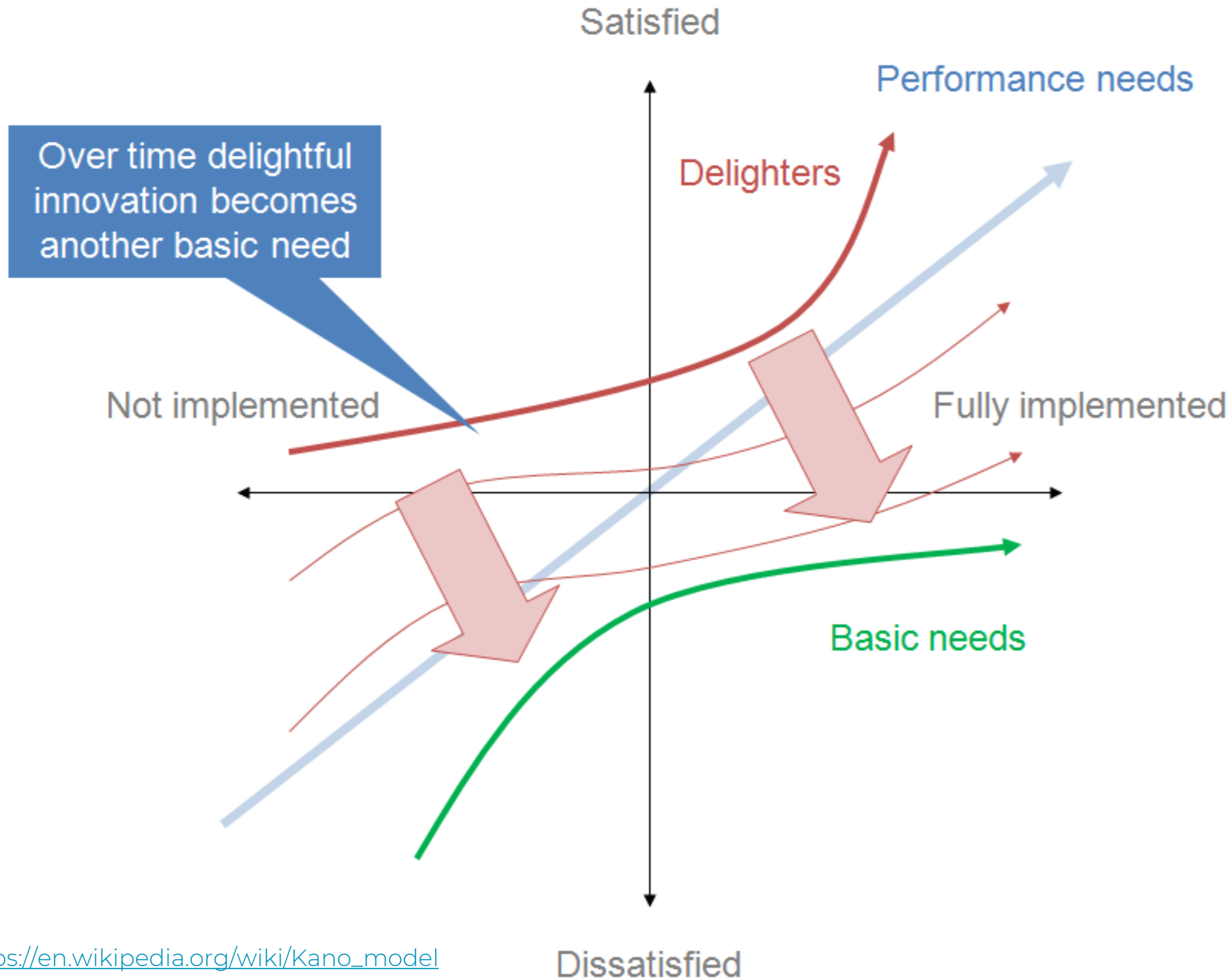
Internal Developer Platform

Agile Practices = Built in Quality

Automation = Fast Feedback

Transparency = Trust

Product development and customer satisfaction - the KANO model

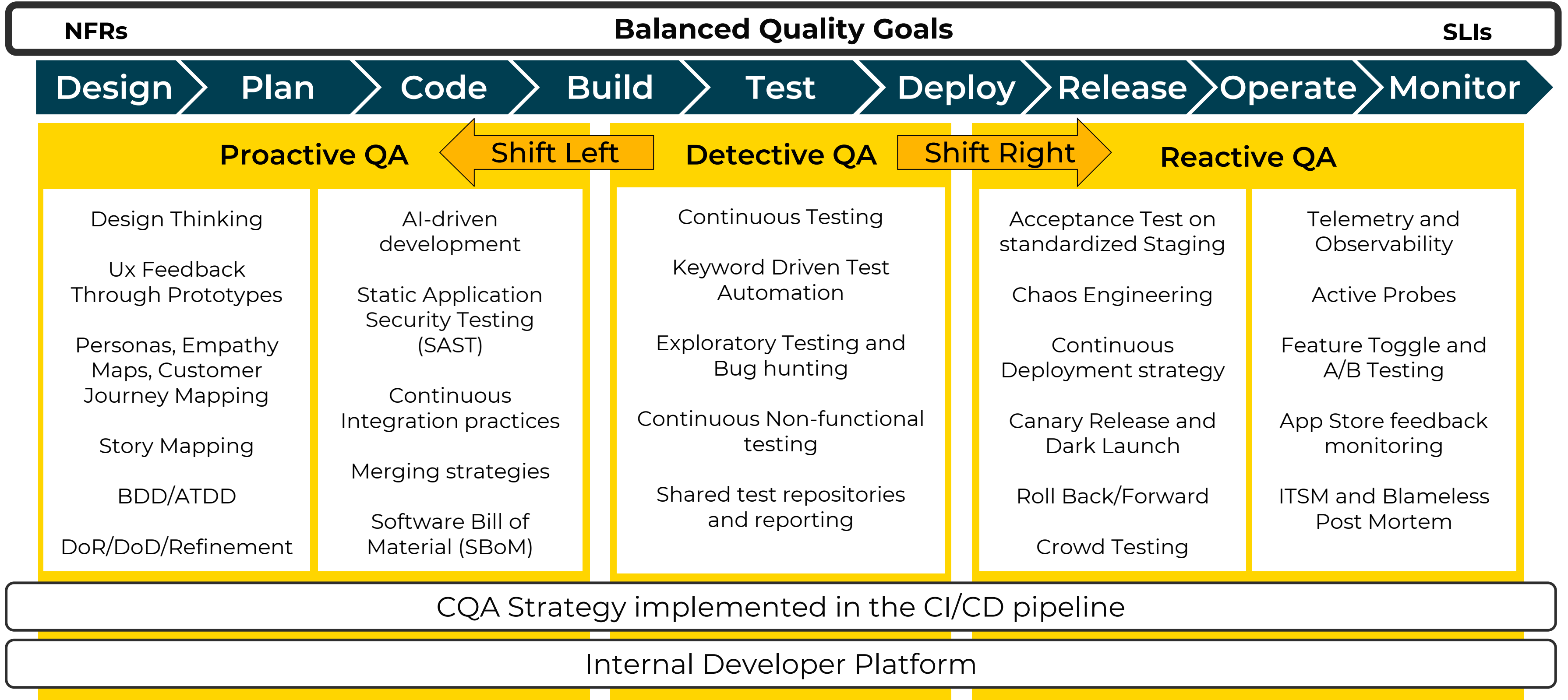


- Must-be Quality
- One-dimensional Quality
- Attractive Quality
- Indifferent Quality
- Reverse Quality

Balanced Quality Goals

<p>USER FRIENDLINESS QA</p> <p>QUALITY ASPECTS</p> <p>Usability Aspect: Will users find the product convenient and easy to work with? But more important: will they enjoy it?</p> <p>▼ +</p>	<p>TESTABILITY QA</p> <p>QUALITY ASPECTS</p> <p>Maintainability Aspect: Everything and anything that impacts how easy or hard your testing efforts are.</p> <p>▼ +</p>	<p>STRUCTURE QA</p> <p>QUALITY ASPECTS</p> <p>Reliability Aspect: The architecture, technologies used, different programming languages and duplicated code.</p> <p>▼ +</p>	<p>STABILITY QA</p> <p>QUALITY ASPECTS</p> <p>Reliability Aspect: Is your product going down every 15 minutes? What could be the root cause?</p> <p>▼ +</p>	<p>SECURITY AND PERMISSION QA</p> <p>QUALITY ASPECTS</p> <p>Reliability Aspect: Watch Dogs, cameras, user credentials, firearms or alarm systems. What's keeping the bad guys out of, and away from your app?</p> <p>▼ +</p>
<p>SCALABILITY QA</p> <p>QUALITY ASPECTS</p> <p>Efficiency Aspect: Hardware is cheap these days. Just enter your credit card and there you go. How much hardware can you juggle?</p> <p>▼ +</p>	<p>SAFE-GUARD QA</p> <p>QUALITY ASPECTS</p> <p>Security Aspect: What's keeping the bad guys out?</p> <p>▼ +</p>	<p>RESOURCE MANAGEMENT QA</p> <p>QUALITY ASPECTS</p> <p>Efficiency Aspect: How does your app handle resources? ? Can they be reduced?</p> <p>▼ +</p>	<p>PERFORMANCE QA</p> <p>QUALITY ASPECTS</p> <p>Efficiency Aspect: How well can your app execute the commands it's been given? How about many commands all at once?</p> <p>▼ +</p>	<p>OPERATIONS QA</p> <p>QUALITY ASPECTS</p> <p>Maintainability Aspect: How easy or hard is it to rectify a problem in production after the product is released?</p> <p>▼ +</p>
<p>OBSERVABILITY QA</p> <p>QUALITY ASPECTS</p> <p>Controllability Aspect: How well can you see what's actually going on within the product? Can you answer new questions without deploying code?</p> <p>▼ +</p>	<p>INTERNATIONALIZATION QA</p> <p>QUALITY ASPECTS</p> <p>Usability Aspect: Currency, time zone, language, right-to-left... Can you use your application anywhere in the world?</p> <p>▼ +</p>	<p>INSTALLABILITY QA</p> <p>QUALITY ASPECTS</p> <p>Portability Aspect: All the factors that matter during the installation process of your app.</p> <p>▼ +</p>	<p>IMPARTIALITY QA</p> <p>QUALITY ASPECTS</p> <p>Functionality Aspect: Software built by humans imitates the social structures that are considered 'normal' by those humans. This includes their biases, favouritism & prejudices.</p> <p>▼ +</p>	<p>FUNCTIONALITY QA</p> <p>QUALITY ASPECTS</p> <p>Functionality Aspect: The user needs to be able to do X, so we test it. What could go wrong?</p> <p>▼ +</p>

Add Quality goals by Product Management



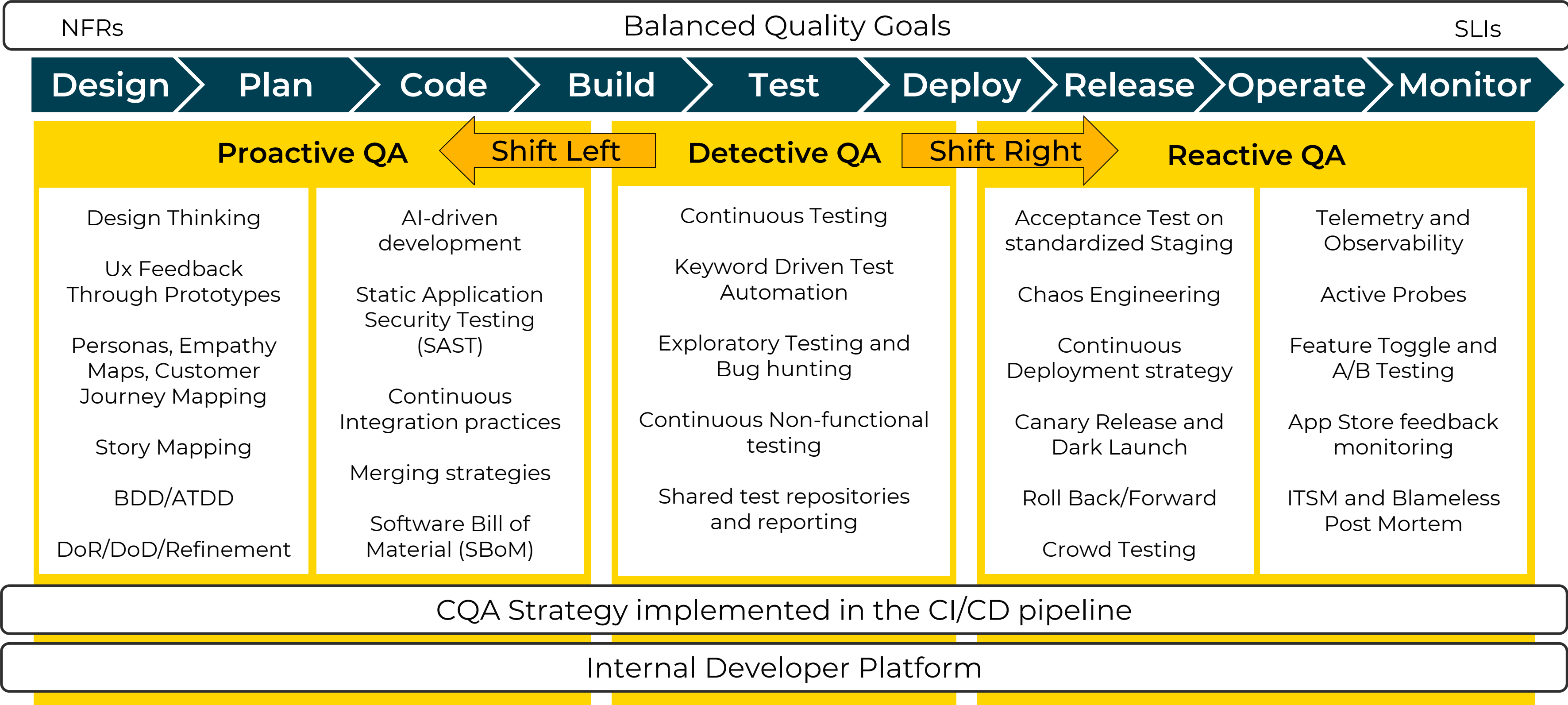
Agile Practices = Built in Quality

Automation = Fast Feedback

Transparency = Trust



Three pillar Balanced Quality model



Agile Practices = Built in Quality

Automation = Fast Feedback

Transparency = Trust



**Nice, but how
to apply?**



Example – User Friendliness

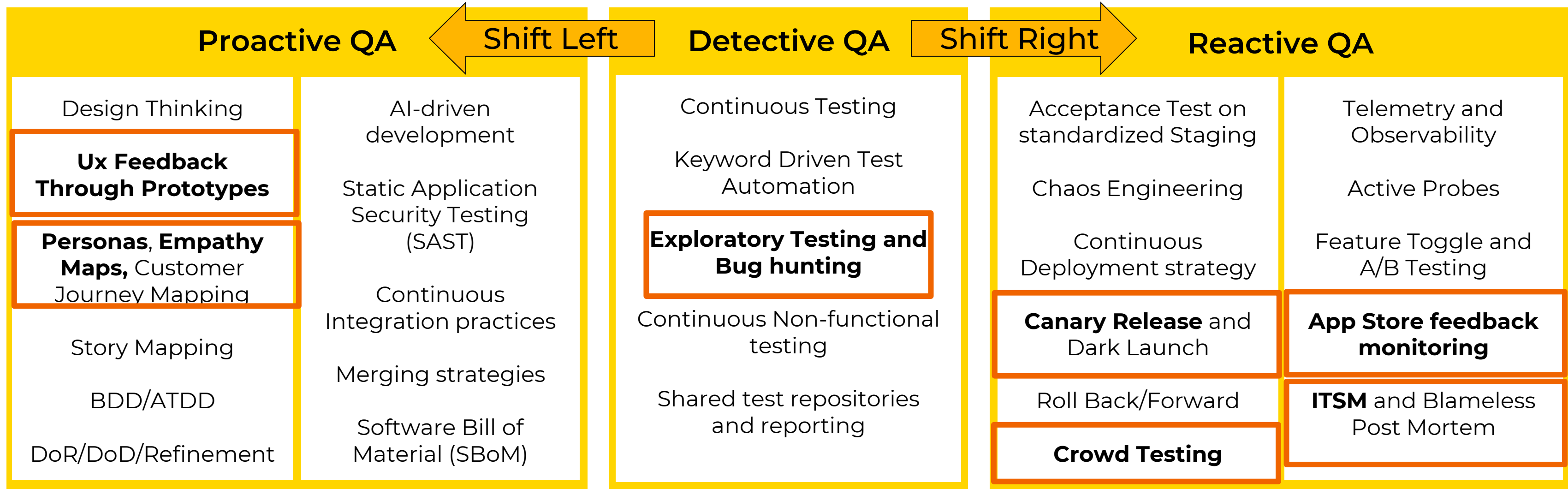
USER FRIENDLINESS QA
 QUALITY ASPECTS

Usability Aspect: Will users find the product convenient and easy to work with? But more important: will they enjoy it?

⌵ +



NFRs Balanced Quality Goals SLIs



CQA Strategy implemented in the CI/CD pipeline

Internal Developer Platform

Agile Practices = Built in Quality

Automation = Fast Feedback

Transparency = Trust

Example – performance



PERFORMANCE QA

QUALITY ASPECTS

Efficiency Aspect: How well can your app execute the commands it's been given? How about many commands all at once?

v
+

NFRs
SLIs
Balanced Quality Goals



Proactive QA	Shift Left	Detective QA	Shift Right	Reactive QA
<ul style="list-style-type: none"> Design Thinking Ux Feedback Through Prototypes Personas, Empathy Maps, Customer Journey Mapping Story Mapping BDD/ATDD DoR/DoD/Refinement 	<ul style="list-style-type: none"> AI-driven development Static Application Security Testing (SAST) <li style="border: 2px solid orange; padding: 5px; text-align: center;">Continuous Integration practices Merging strategies Software Bill of Material (SBoM) 	<ul style="list-style-type: none"> Continuous Testing Keyword Driven Test Automation Exploratory Testing and Bug hunting <li style="border: 2px solid orange; padding: 5px; text-align: center;">Continuous Non-functional testing Shared test repositories and reporting 	<ul style="list-style-type: none"> <li style="border: 2px solid orange; padding: 5px; text-align: center;">Acceptance Test on standardized Staging Chaos Engineering Continuous Deployment strategy <li style="border: 2px solid orange; padding: 5px; text-align: center;">Canary Release and Dark Launch Roll Back/Forward Crowd Testing 	<ul style="list-style-type: none"> Telemetry and Observability <li style="border: 2px solid orange; padding: 5px; text-align: center;">Active Probes Feature Toggle and A/B Testing App Store feedback monitoring ITSM and Blameless Post Mortem

CQA Strategy implemented in the CI/CD pipeline

Internal Developer Platform

Agile Practices = Built in Quality

Automation = Fast Feedback

Transparency = Trust

Summary



Continuous Testing is important however...

...Quality shall be handled on the Left and Right, with a balance

Internal Development Platform is your foundation to build on

Leadership to set Balanced Quality goals and follow it up

DevOps needs even more Quality



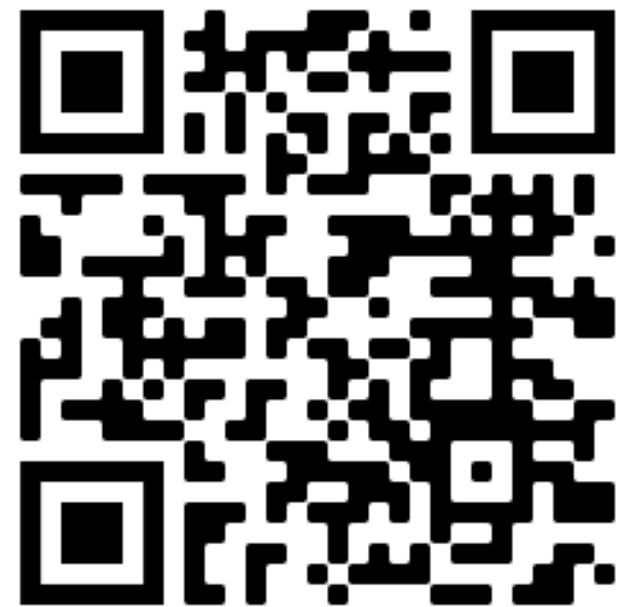
Questions?

Thank you

eficode

 www.linkedin.com/in/szellszilard/

<https://www.eficode.com/szilard-szell>



For more info, visit www.eficode.com

