How to DevOps your Testing Strategy

An Exercise in Value Stream Analysis

Adam Auerbach

Adam Auerbach

Head of Cloud Agility and Testing Practice

Adam Auerbach

VP, Cloud Agility and Testing

Joined EPAM in April of 2018 as VP, DevTestSecOps, based in Newtown PA.

Prior to EPAM, Adam served as the VP of Quality and DevOps Engineering at Lincoln Financial Group, where he was responsible for leading the implementation of Continuous Testing and Continuous Delivery.

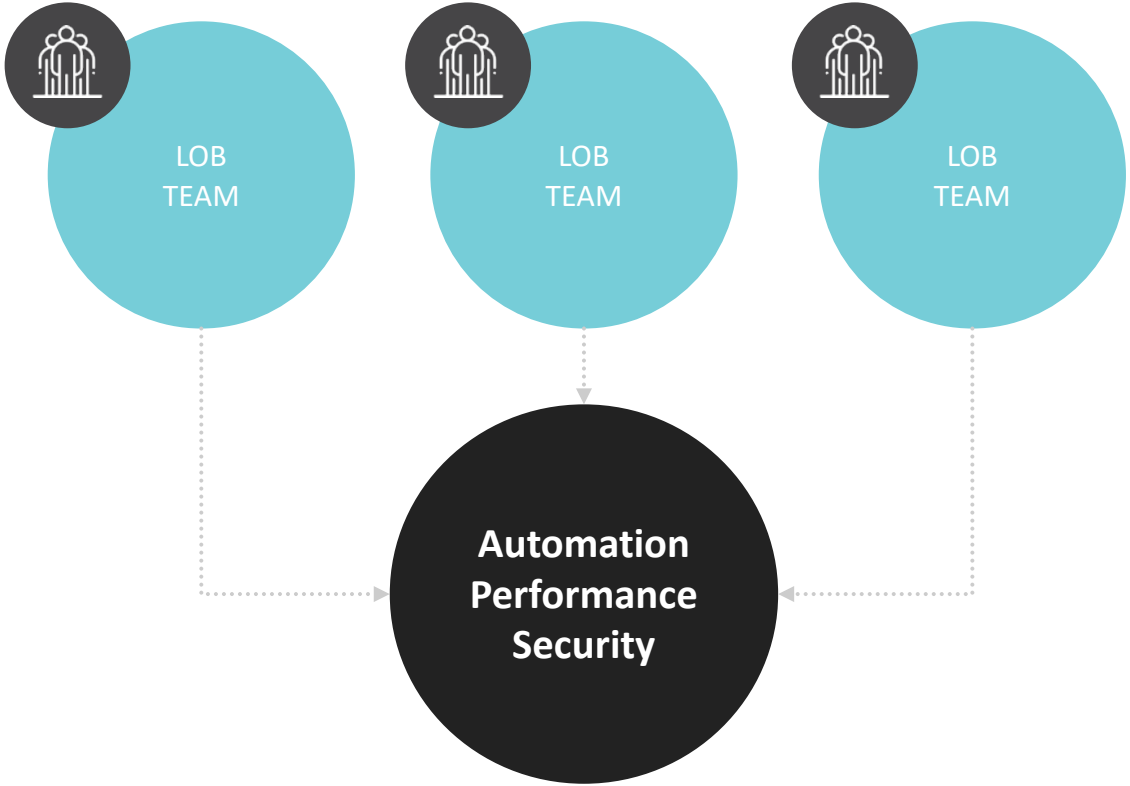
Adam regularly speaks at conferences as well as writing blogs and articles on Testing and DevOps.

You can find more of Adam's thinking on the TechWell blog or follow him on Twitter at @bugman31.

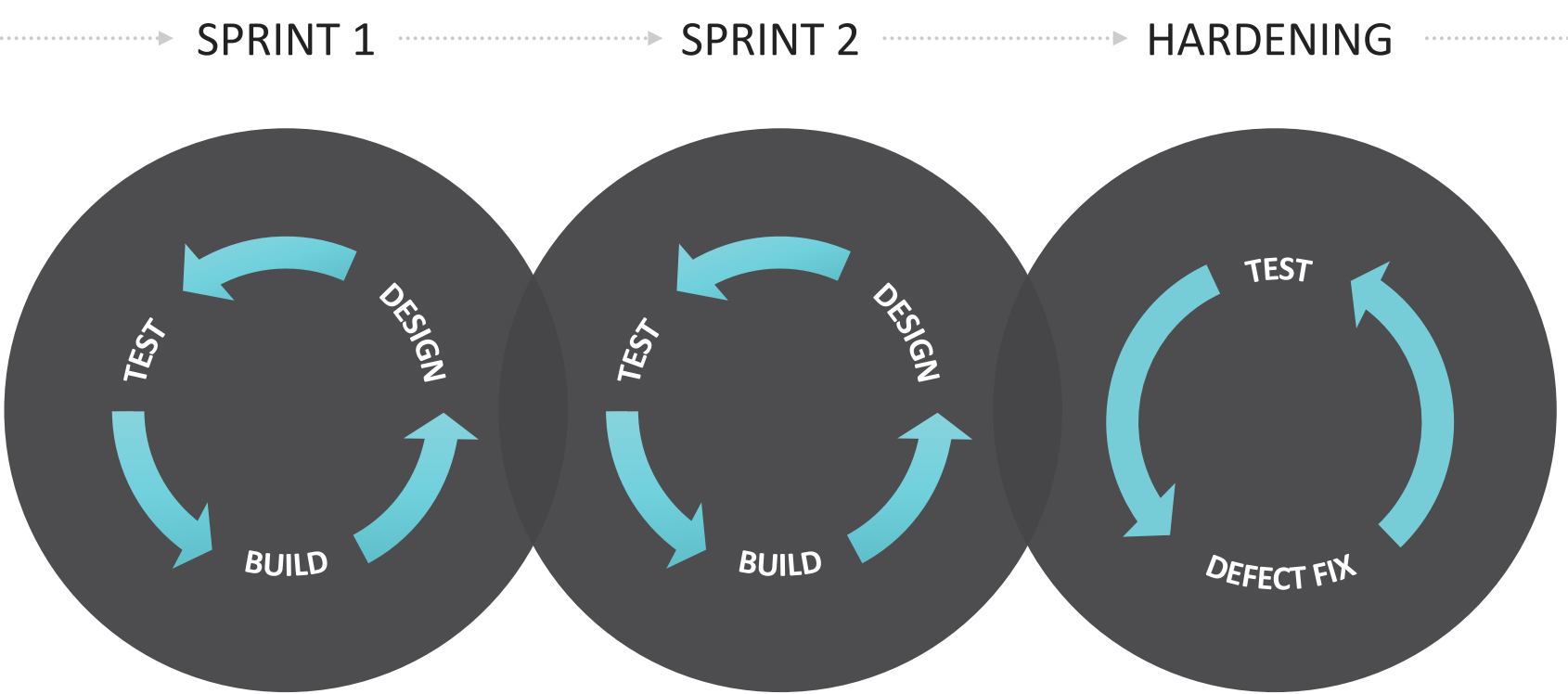
©2018 EPAM Systems, Inc.

AGILE

Traditional QA Organization



Agile?


SPRINT 1
SPRINT 2
HARDENING
TEST
DESIGN
BUILD
TEST
DESIGN
BUILD
TEST
DEFECT FIX

DEVOPS

DevOps is a Philosophy

DELIVER HIGH QUALITY WORKING SOFTWARE FASTER

Agile Pod
(Dev, QA, PO, BSA)

+

Infrastructure

Arch.

Prod Support

Shared Services (e.g. Security Testing, Perf Testing)

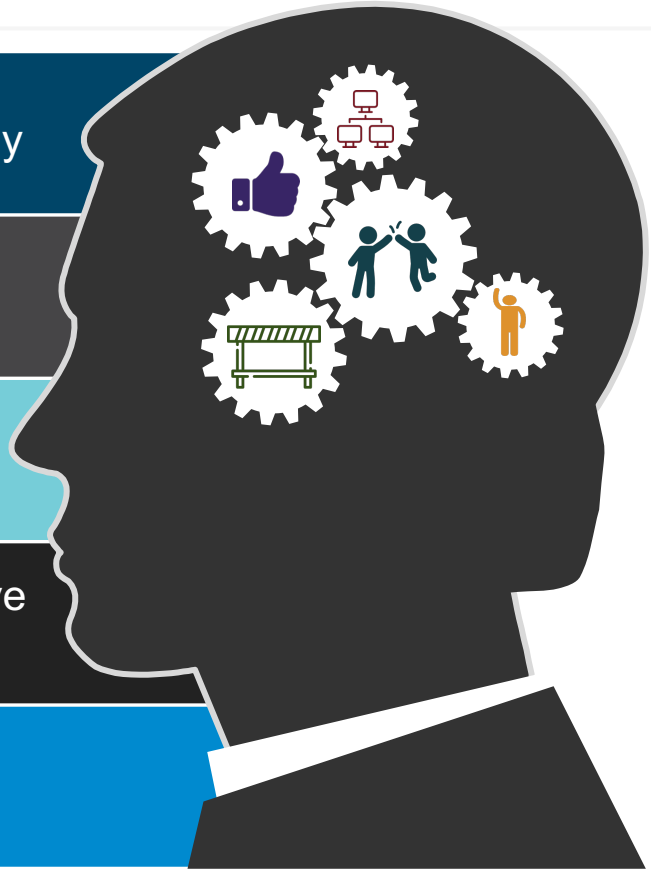
DevOps is a philosophy where teams are accountable for everything required to get their code developed, tested and deployed to production, while shared service teams provide the automation and tools to enable them.

"You Build it You Own it"

BUGMAN31

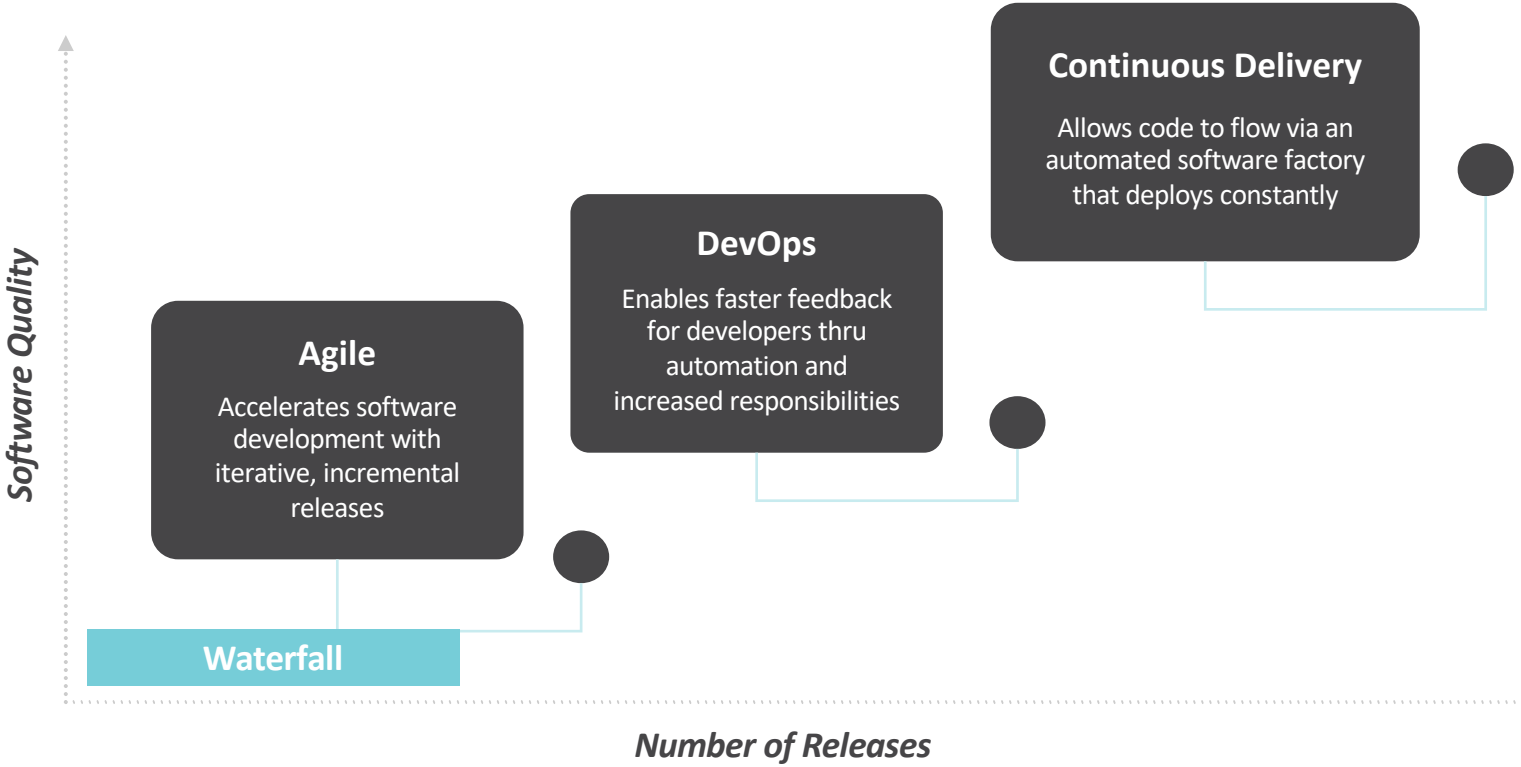
5 Key Principles to DevOps

1. Build once, deploy many
2. Enable self-service and on-demand capabilities
3. Leverage SMEs for support and best practices
4. Identify opportunities to remove bottlenecks
5. Increase and assume accountability



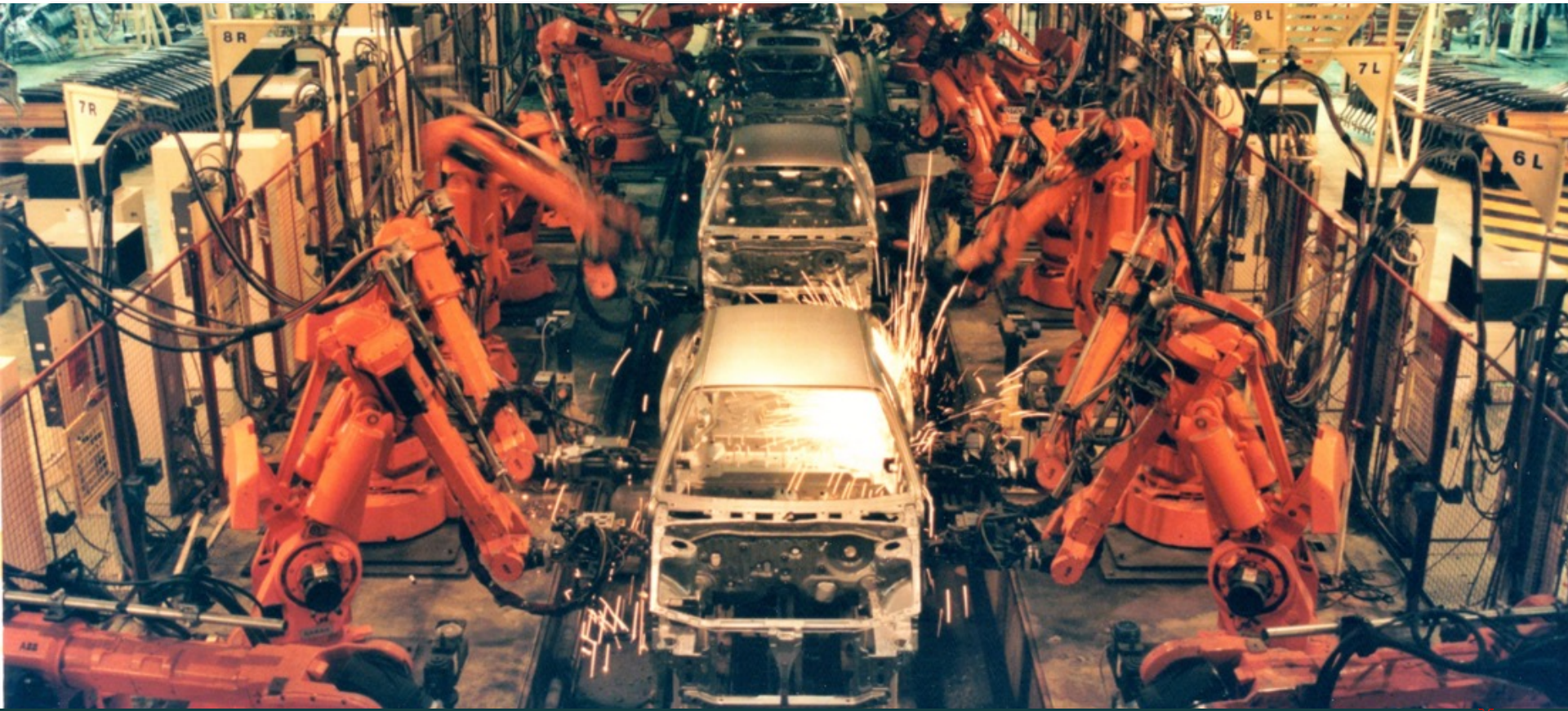
BUGMAN31

DevOps is the Next Part of Your Agile Journey



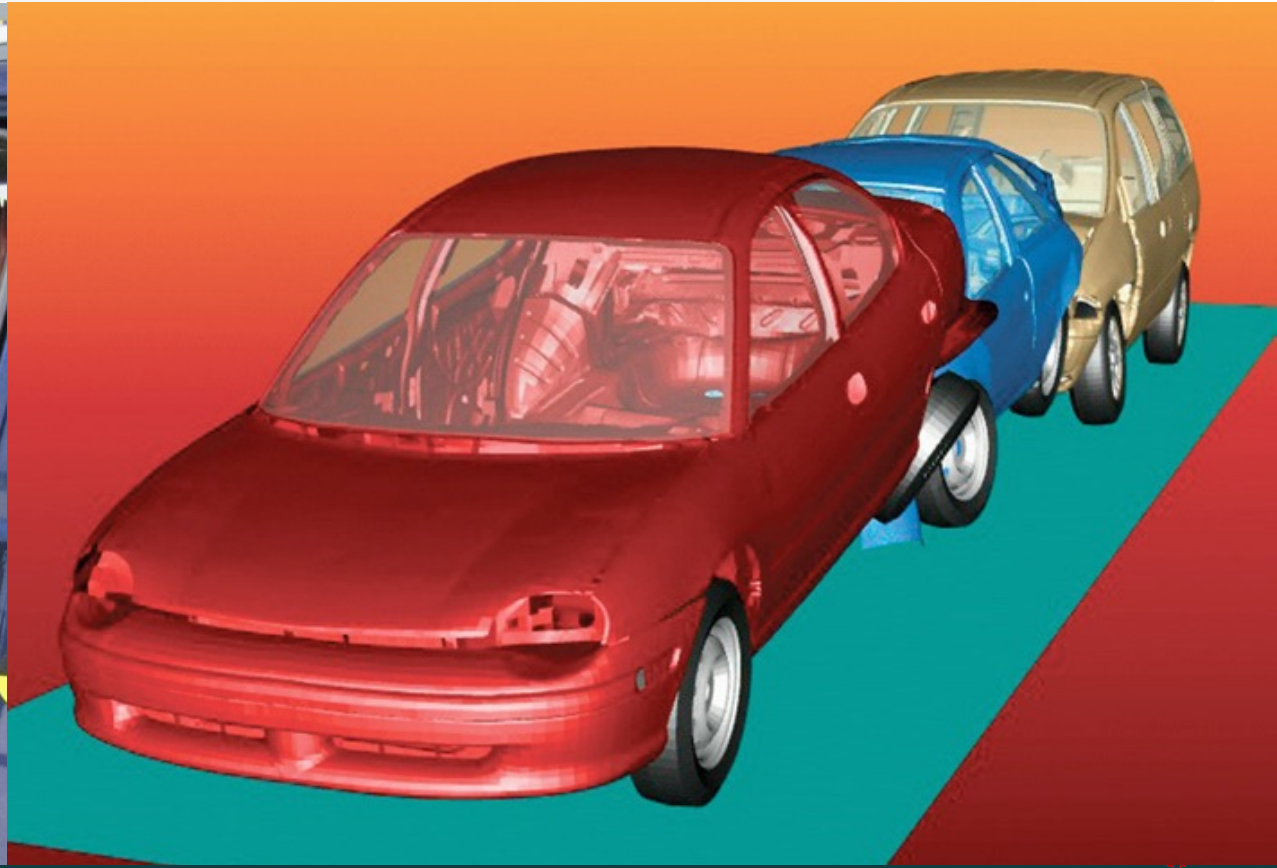
What is DevOps?

Automate Everything



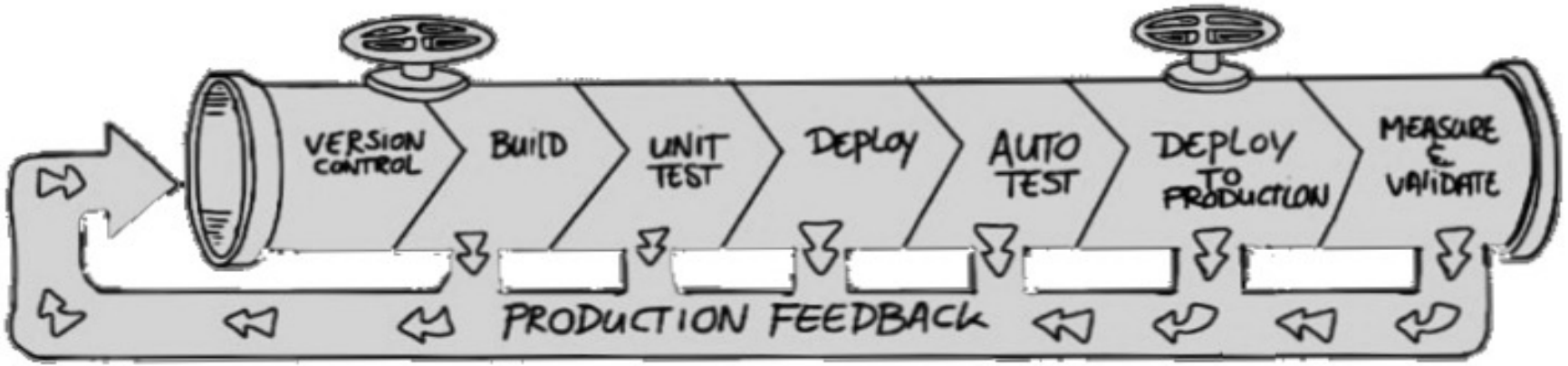
Fast Feedback

Remove Constraints

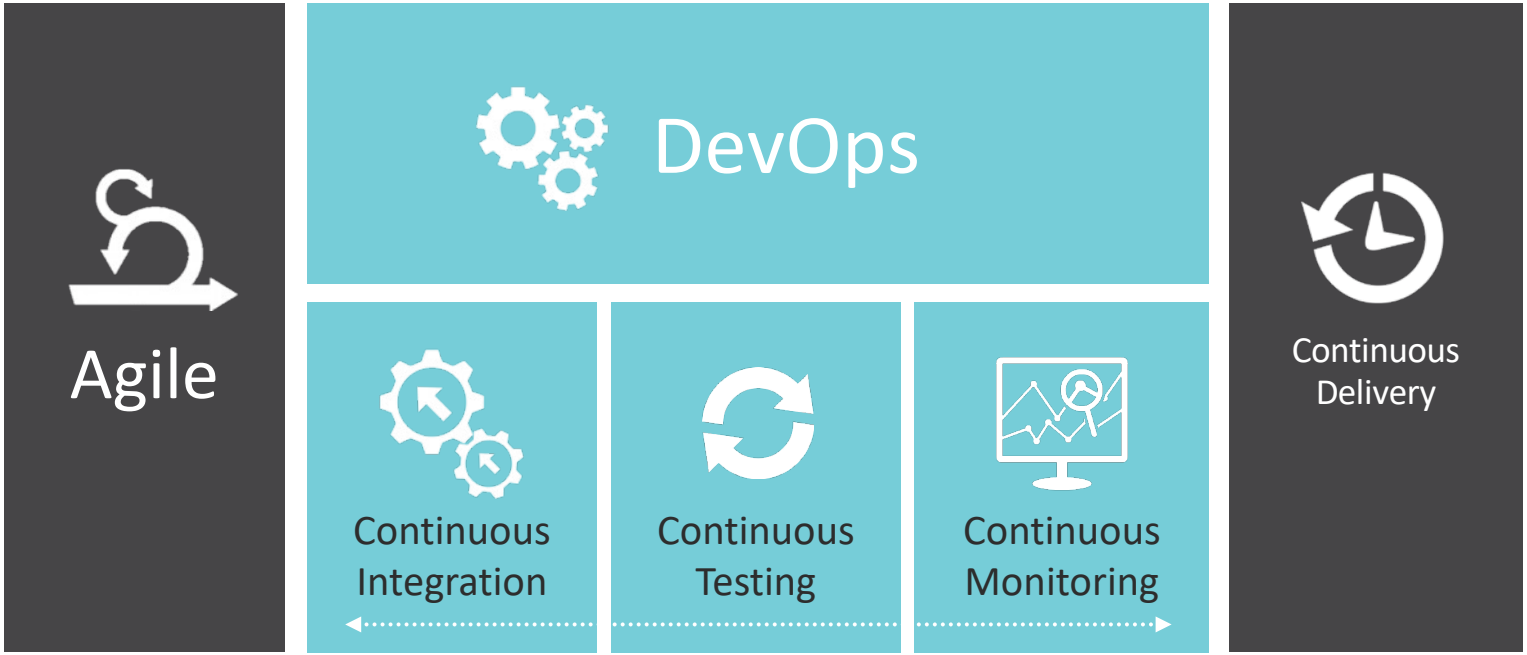


Collaboration Early and Often

Focus on Flow



Agile + DevOps



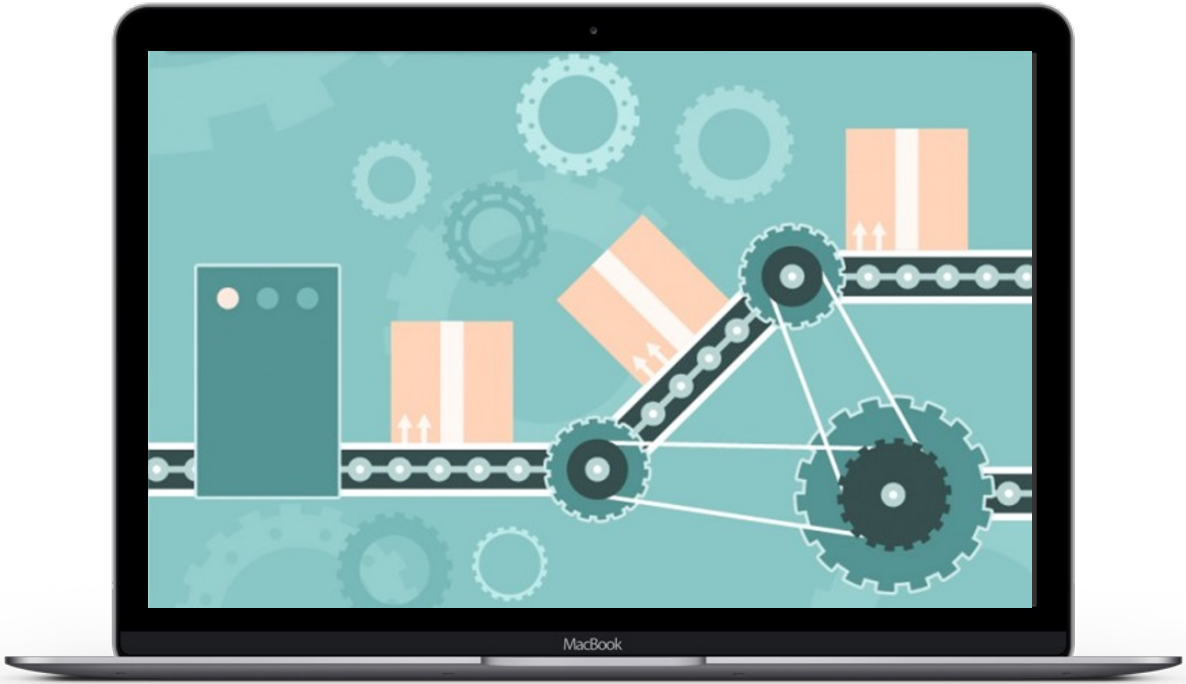
Value Stream Analysis – Setting Context

1. Pick Case Studies – At least 4

2. Set Context (15min)
    - Company and Product
    - Agile or Waterfall?
    - Describe team's makeup
    - Overall release timeline (sprint length, time to test, deploy, releasee)
    - Describe "Quality" of product (defect leakage, test fail rate, pass rate)

HOW DOES OUR TESTING HAVE TO CHANGE?

Pipelines



Real-Time Automation

BEHAVIOR-DRIVEN DEVELOPMENT (BDD)

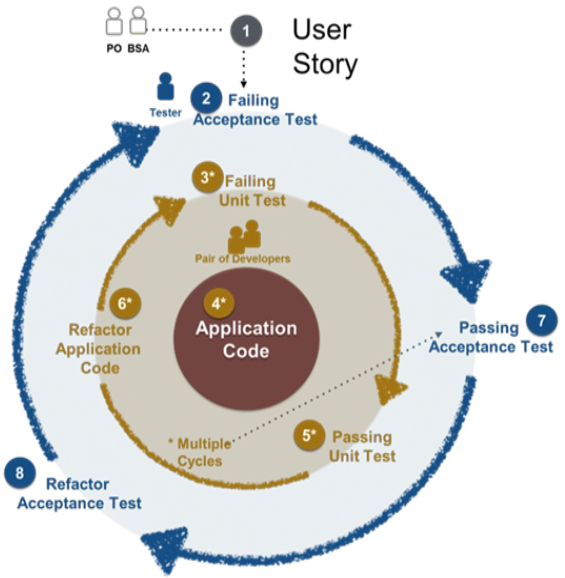
- Team Focused
- Stress on Stories
- Add a Test
- Run all Tests
- Write Code
- Refactor Until Added Test Passes

TEST-DRIVEN DEVELOPMENT (TDD)

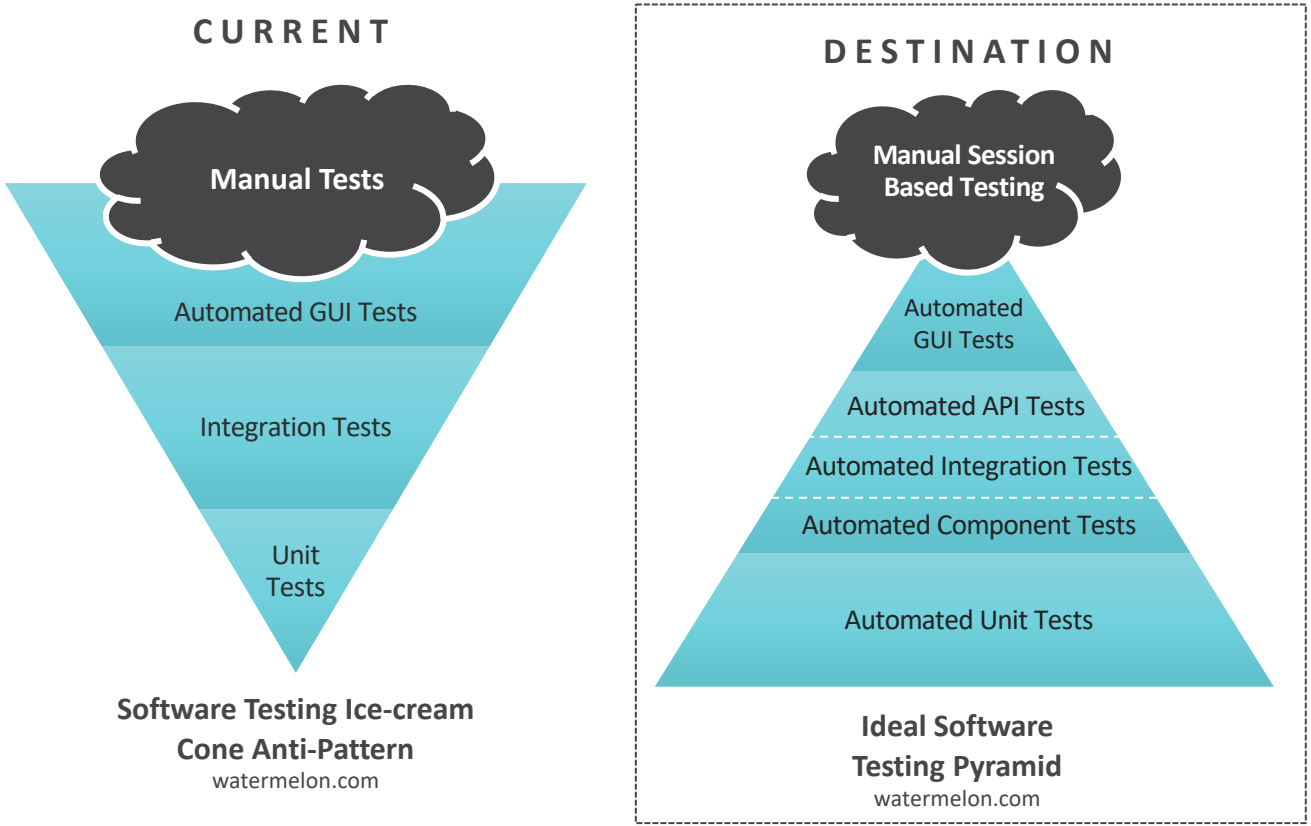
- Developer Focused
- Closer to Unit Level
- Add a Test
- Run all Tests
- Write Code
- Refactor Until Added Test Passes

ACCEPTANCE TEST-DRIVEN DEVELOPMENT (ATDD)

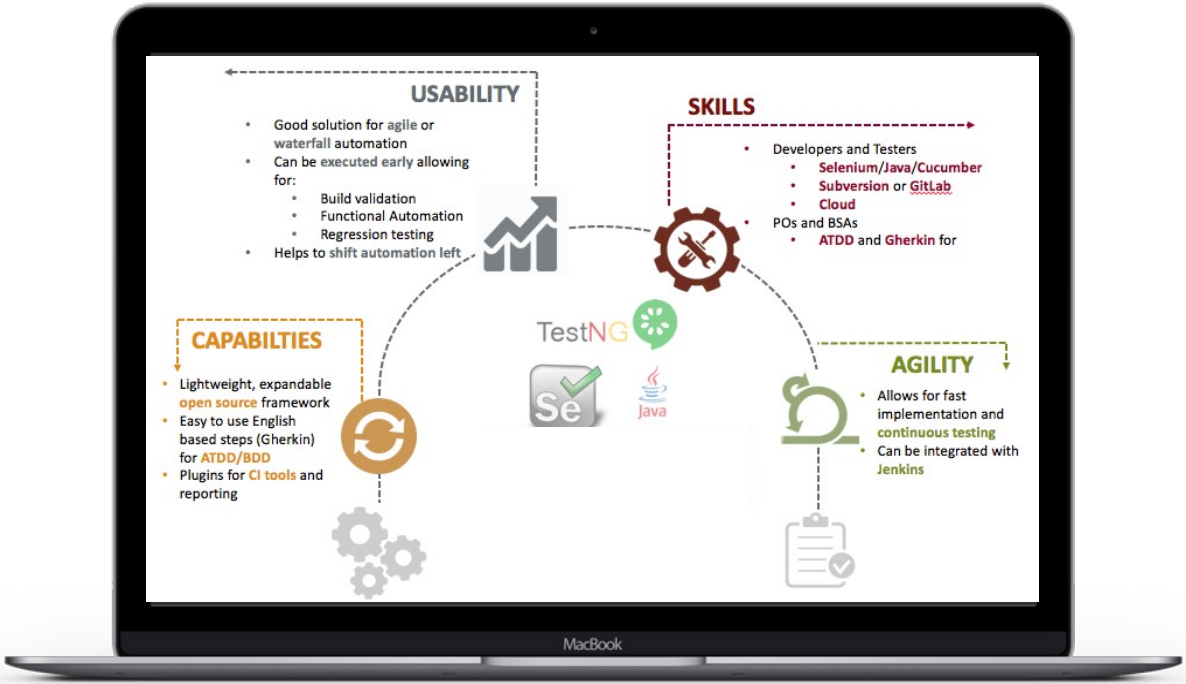
- Team Focused
- Stress on Acceptance Criteria
- Add a Test
- Run all Tests
- Write Code
- Refractor Until Added Test Passes



Testing Approach

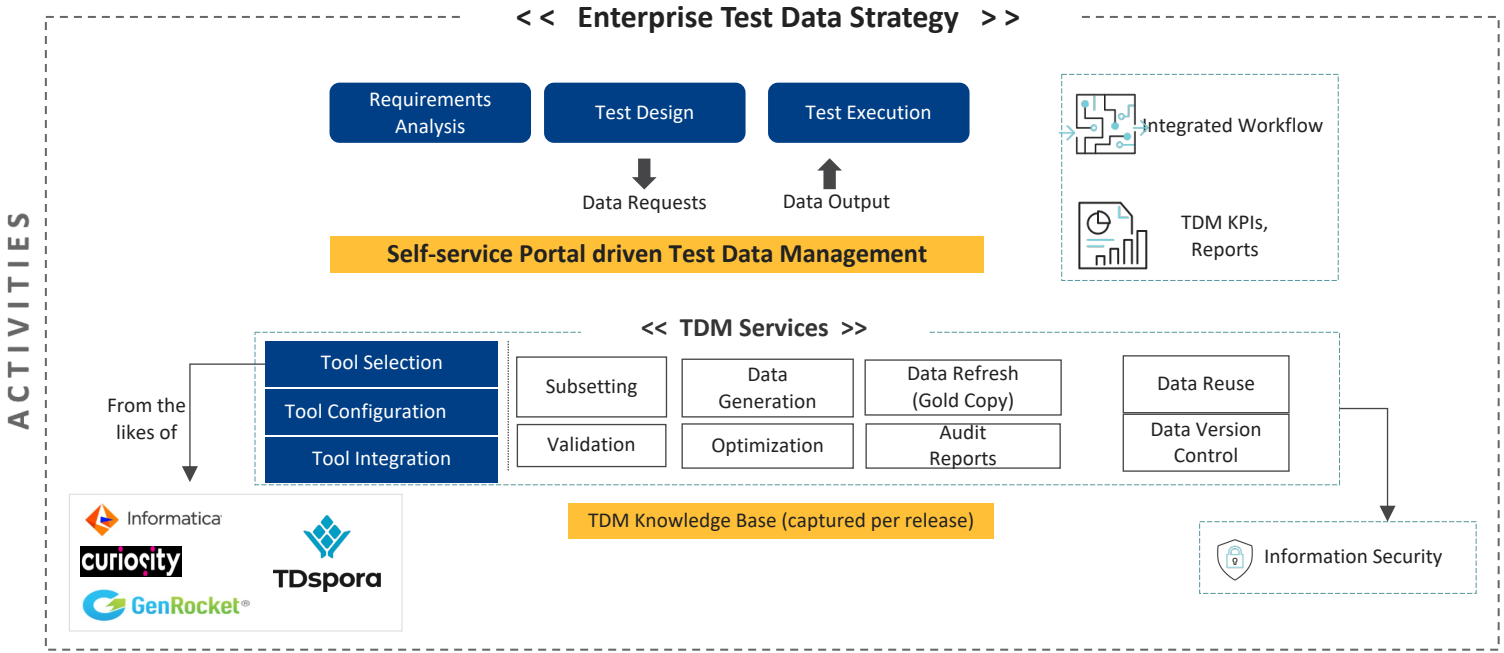


Open Source Tools



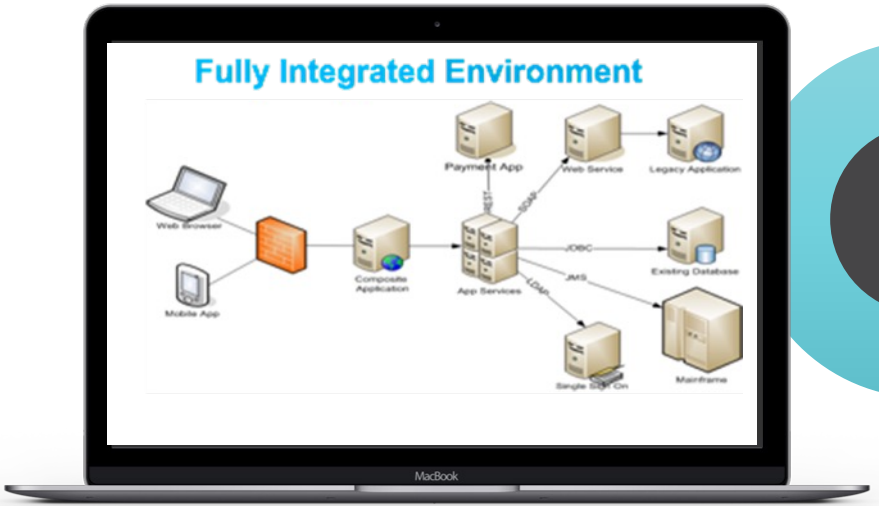
Environments

Test Data Management



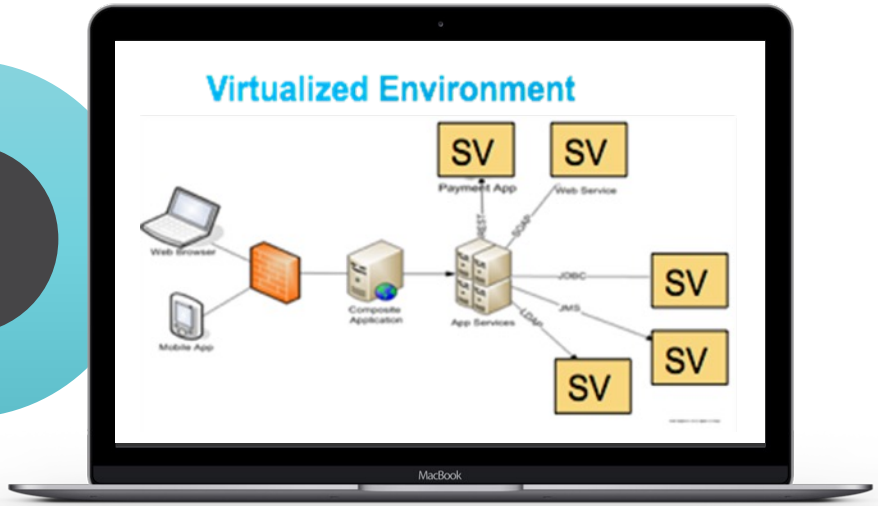
Service Virtualization

INTEGRATED ENVIRONMENT



VS.

VIRTUALIZED ENVIRONMENT



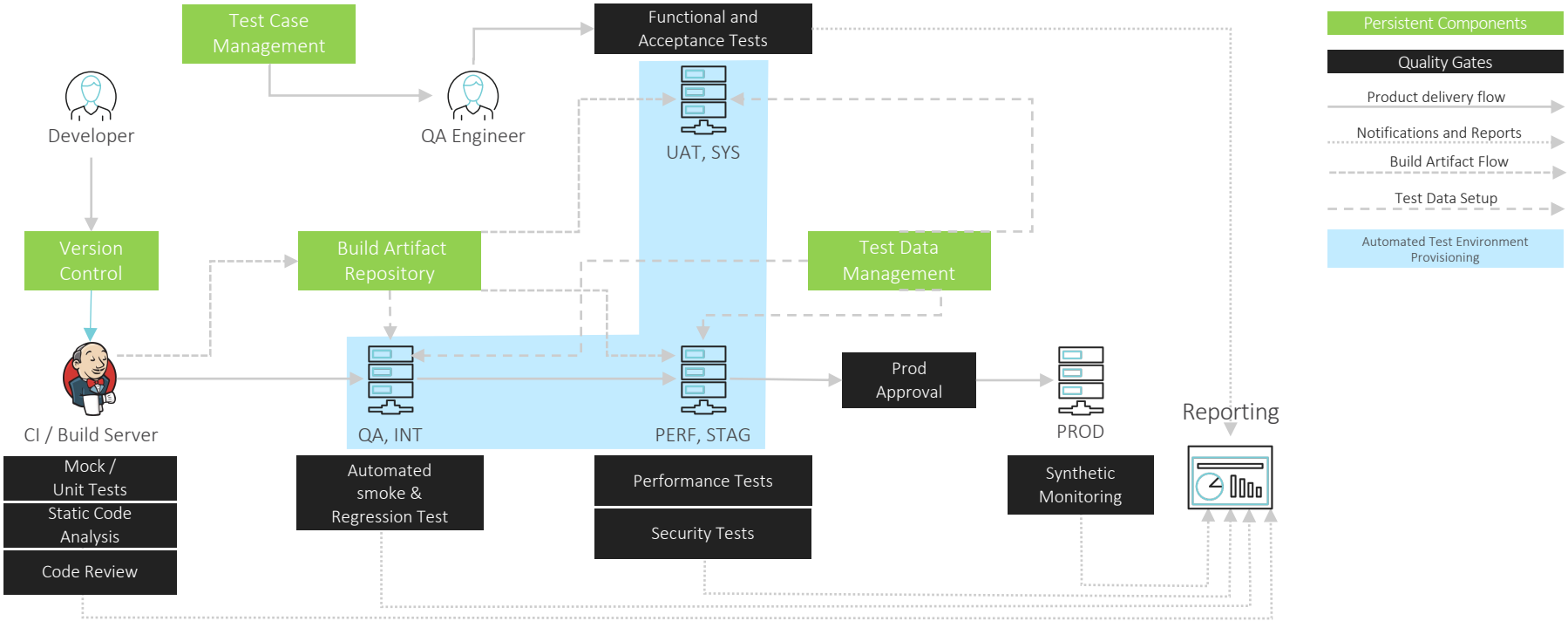
Realtime Insights

Reportportal.io

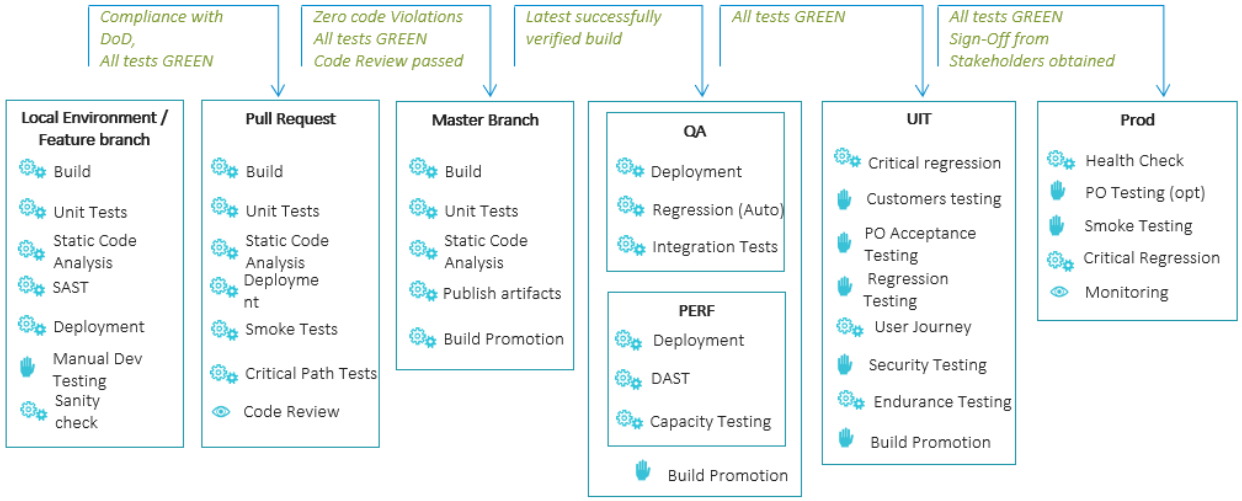
CapitalOne.io

Continuous Testing

Delivery and Governance are tracked and supported by KPIs and industry standard quality gates in every level of testing architecture.



Example of a Continuous Testing Pipeline



HIGHLIGHTS

- Automated quality gate is implemented as a step in the pipeline that outputs a binary result (pass/fail)
- Based on the result the decision on whether the build can move further down the pipeline is made automatically

EXPECTED BENEFITS

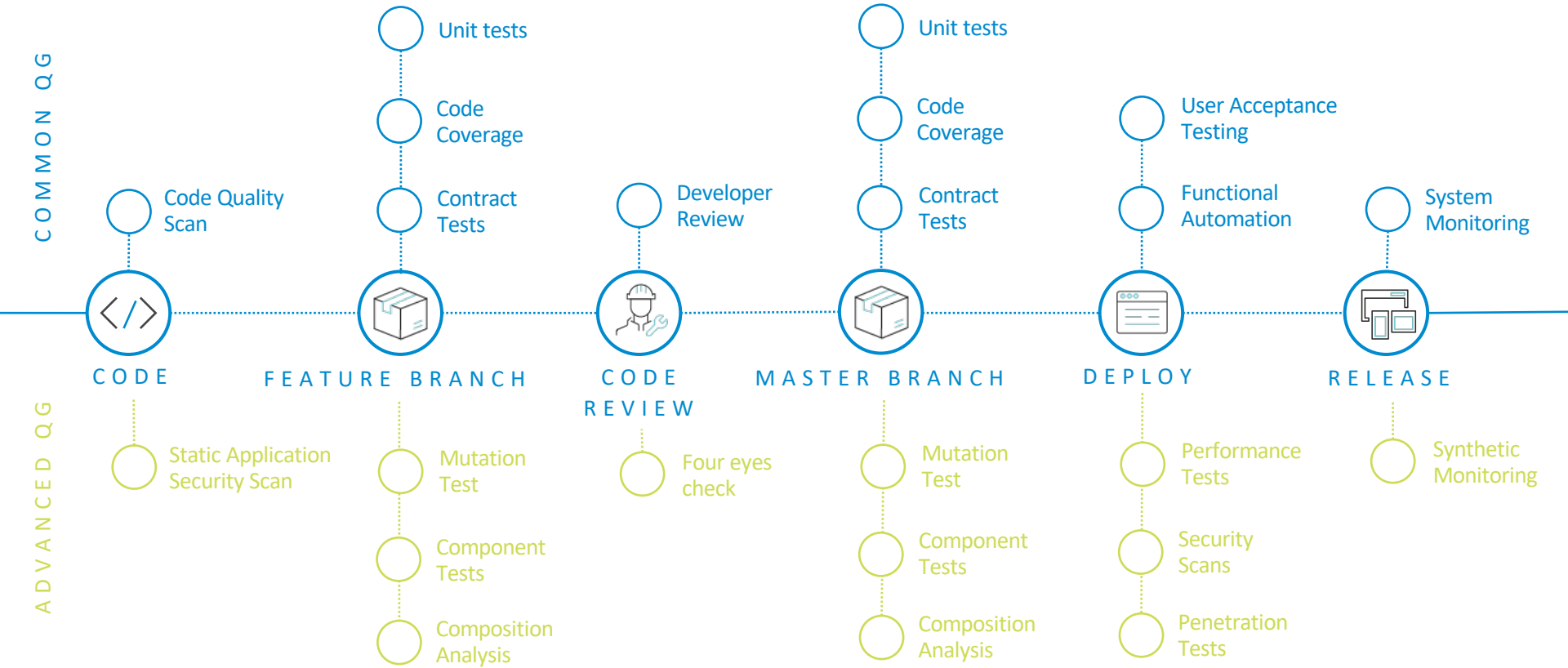
- Time savings on build/deployment administration
- Time savings on test result analysis
- Harmful code remains at hands of the author until fixed and doesn't affect the rest of the team
- Tests are properly maintained

KPIs TO MEASURE VALUE

- Count of bugs let through the gates
- Count of bug-free code increments

epam | CONFIDENTIAL. Copyright © 2019

Continuous Testing Quality Gates



Value Stream Analysis – Setting Context

1. Describe Testing (15min)
   - How much is manual vs Automated?
   - What type of automation is done?
   - What tools are used?
   - Who does the automation?
   - What is the CI process? Is Automation part of it?

Continuous Testing isn't common yet

CT standards:

- Code review, code standards, pair programming
- Pre-commit verification, automated deployment process
- Automation tests as a part of DoD.
- Automated quality gates, quality metrics and dashboards covering all test stages, risk-based quality management
- Self-service test data management and service virtualization
- Full traceability between requirements, tests and test results
- Non-functional requirements are assessed continuously
- Continuous improvement process.
- Real-time app monitoring
- Self-service infrastructure

Test process

25%(2) have defect containment level higher than 95%	79%(1)(2) have exploratory testing	90%(2) have test cases combined into suites	29%(2) run performance tests daily or per sprint	12%(1) run security tests daily or per sprint

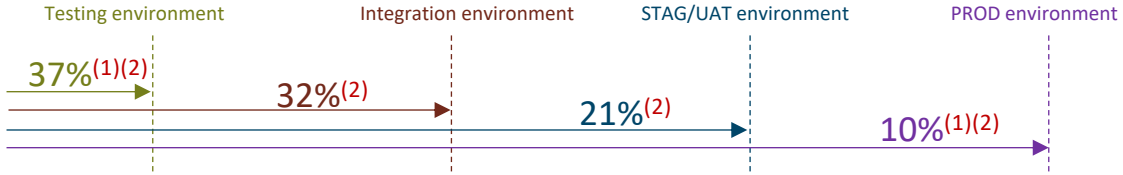
Automation health

35%(2) have unit tests Coverage > 80%	70%(1) have smoke and regression tests automated	38%(2) have regression tests coverage > 80%	24%(1) have > 80% of regression tests on non-UI level	55%(1) have Pass Ratio > 90%

Delivery health

37%(2) have resolution time for blocking issues < 1day	52%(1) have automation as a part of DoD	43%(1)(2) have ability to promote a new feature within 1 week	41%(1)(2) have static code analysis as a quality gate (QG)	42%(1)(2) have smoke test as a QG	33%(2) have regression tests as a QG

Builds promoted automatically to upper environments



Embedded

Software & Hi-Tech (1)

Retail (2)

OPPORTUNISTIC — DEVELOPING — ENGAGING — SUSTAINABLE — TRANSFORMATIVE

Continuous Testing Maturity Model

We think of Continuous Testing as the **Target State of testing**.

Thus, at EPAM we pursue Continuous Testing transformation for our Clients to drive increasing the maturity of Continuous Testing across project teams

OPPORTUNISTIC	DEVELOPING	ENGAGING	SUSTAINABLE	TRANSFORMATIVE
• Manual testing • Perceptional code quality • Defects are not registered • Manual ad hoc testing • Testing process is not repeatable • No testing documentation	• Major cross-dependencies & complex regression • Some quality measurements • Defect management process • Test processes & documentation • Quality of requirements under control • Repeatable test process • Reactive non-functional testing • Low level of **automation**	• Documented and regularly updated test strategy • Quality gates appearing, but not followed • **Automated** regression • **Automation** tests separated by layers • **Automated** non-functional testing • **TA framework** corresponds to best practices • **Automatically** measured quality metrics	• Online dashboards fully cover testing & quality • Test strategy focused on defects prevention • Documented & reused testing practices • **Automated** quality gates • **Automation** testing is main practice • Modern testing service virtualization • **Automated** test data management (TDM) • Test **automation** code is a part of solution & has own quality gates	• BDD is a SSOT about application behavior for the team • Established data-driven continuous Improvement process • Constant 0 bug production quality • AI & ML applied for quality analysis and optimization • Self-service TDM & virtualization

Slow — Time to Market — Fast

Low — Quality of Product & Testing Process — High

CONFIDENTIAL. Copyright ©

Value Stream Analysis – Improvements and Recommendations

1. How do you think it could be improved and why? (15min)
    - Process, Tools, People, etc

SDLC maturity drives immediate needs

Provide a Testing Transformation Journey Tailored to Client's Maturity

Types of Testing Engagements

Waterfall/Agile

Cost Savings thru improved productivity and automation

- Improving automation effectiveness
- Implementing testing best practices
- Maximizing utilization of on/offshore resources

Organizational Needs:

- QA assessment and consulting
- Open source testing frameworks
- AI/ML based tools
- Testing as a Service (manual and automation)
- Onsite, nearshore and offshore

DevOps

Increased value by improving overall quality and time to market

- Continuous quality
- Testers who are engineers
- Non-functional testing is automated
- Quality gates are enforced

Organizational Needs:

- EngX assessment and consulting
- Off-the-shelf performance and security testing platform
- Testing as a Service (performance and security)
- Test data management as service
- Mobile and IoT device farms

Continuous Delivery

Leveraging emerging technologies to know what tests to run and which data to use

- Cloud native automation frameworks and tools
- AI based testing insights
- ML based risk models

Organizational Needs:

- Continuum consulting
- Cloud native CI/CD and testing framework
- Gig workforce
- AI based tools
- ML based data modeling IP

Maturity

Low → *High*

epam | CONFIDENTIAL. Copyright © 2019

Keep QA or not?

QA Organization	Developers Test only
• Test Engineers and SDETs build robust test frameworks, testing solutions are fully embedded in release process for CI/CD • Quick regression, quick feedback	• Developers test their own code from within the code. Tests are in sprint. • Focus is on the system, not on the stakeholders. Tend to miss big picture issues like usability or End to End integration.
• Testers focus on test design and coverage; developers write the automated frameworks • Regression tests may be slow, unautomated scenarios are tested manually	• Developers or project team run test scripts when available.
• Focus is on the stakeholders and value of the product. Exploratory • Slow release cycles, regression testing bottlenecks	• Developers might have some unit tests, Product owners, project managers, other staff does UAT in their spare time. • Shallow focus is on the stakeholders, not the system • High risk

Suggested Core Metrics To Start With

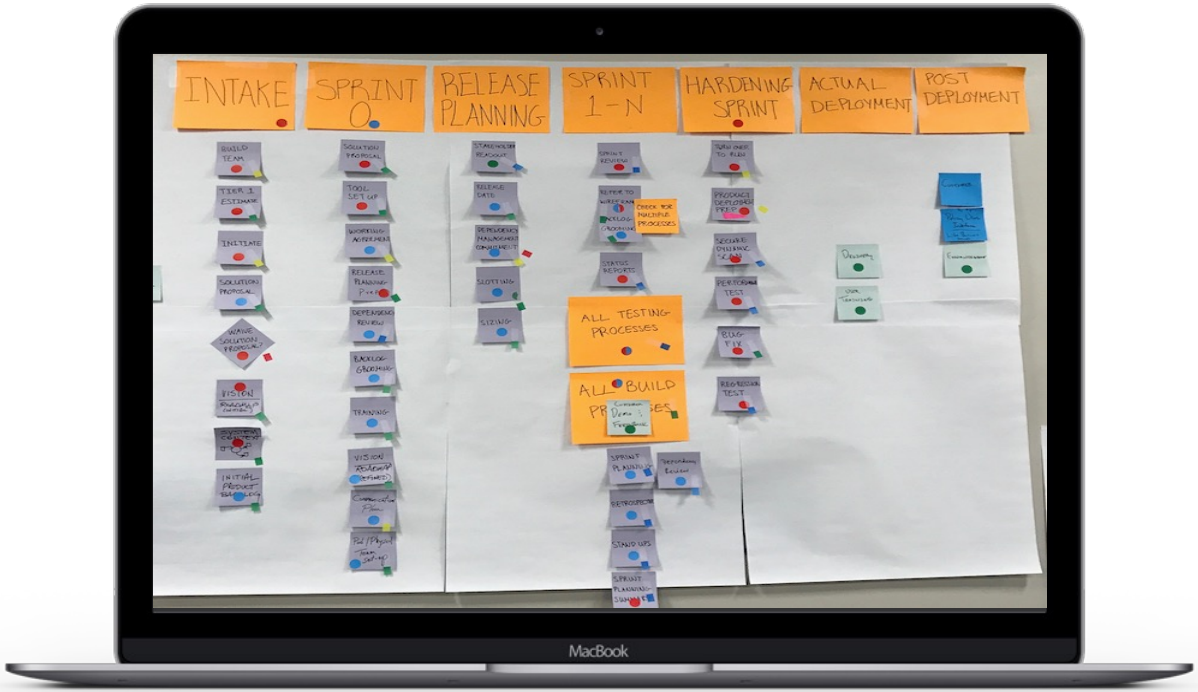
QUALITY METRICS	
METRIC	**TARGET**
Defect containment efficiency	>95%
Number of open defects	Trend not growing
Test coverage	Meet the target
Defects density	Trend not negative
Bug fixing projection	Trend not negative
Defect rejection rate	<10%
Reopened issues rate	<10%
Requirement traceability	100%
Defects age (by priorities)	Top priorities <3 d; All priorities <30 d
Cost of poor quality	<15%

TEST AUTOMATION METRICS	
METRIC	**TARGET**
Execution frequency	Increase
% of results analyzed	100%
% of product issues (Test Automation Stability)	>90%
Execution Time	Custom
Auto test development velocity	Increasing
TA Maintenance effort	<15%
Average Test Success Ratio	Trend not negative
Regression suite Automation Coverage	80%
Smoke suite Automation Coverage	100%

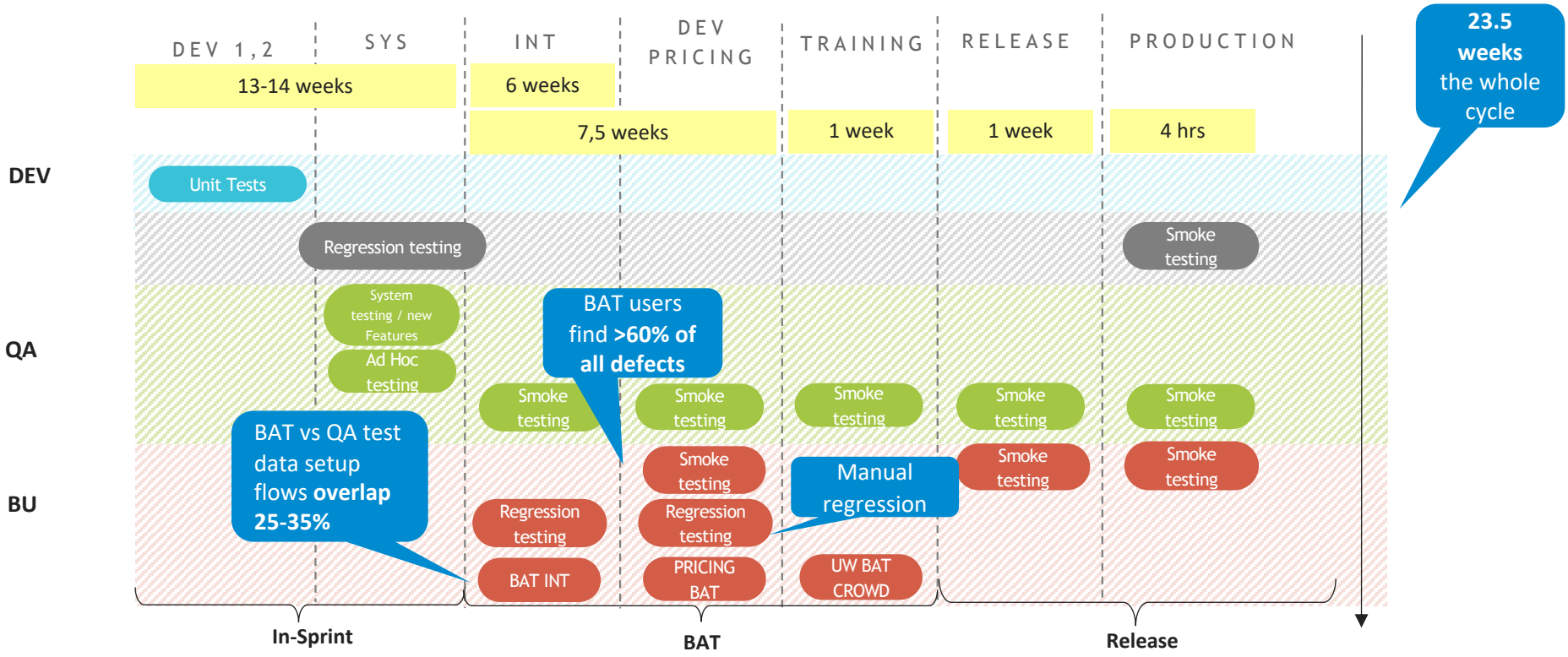
CONFIDENTIAL. Copyright ©

HOW DO YOU START?

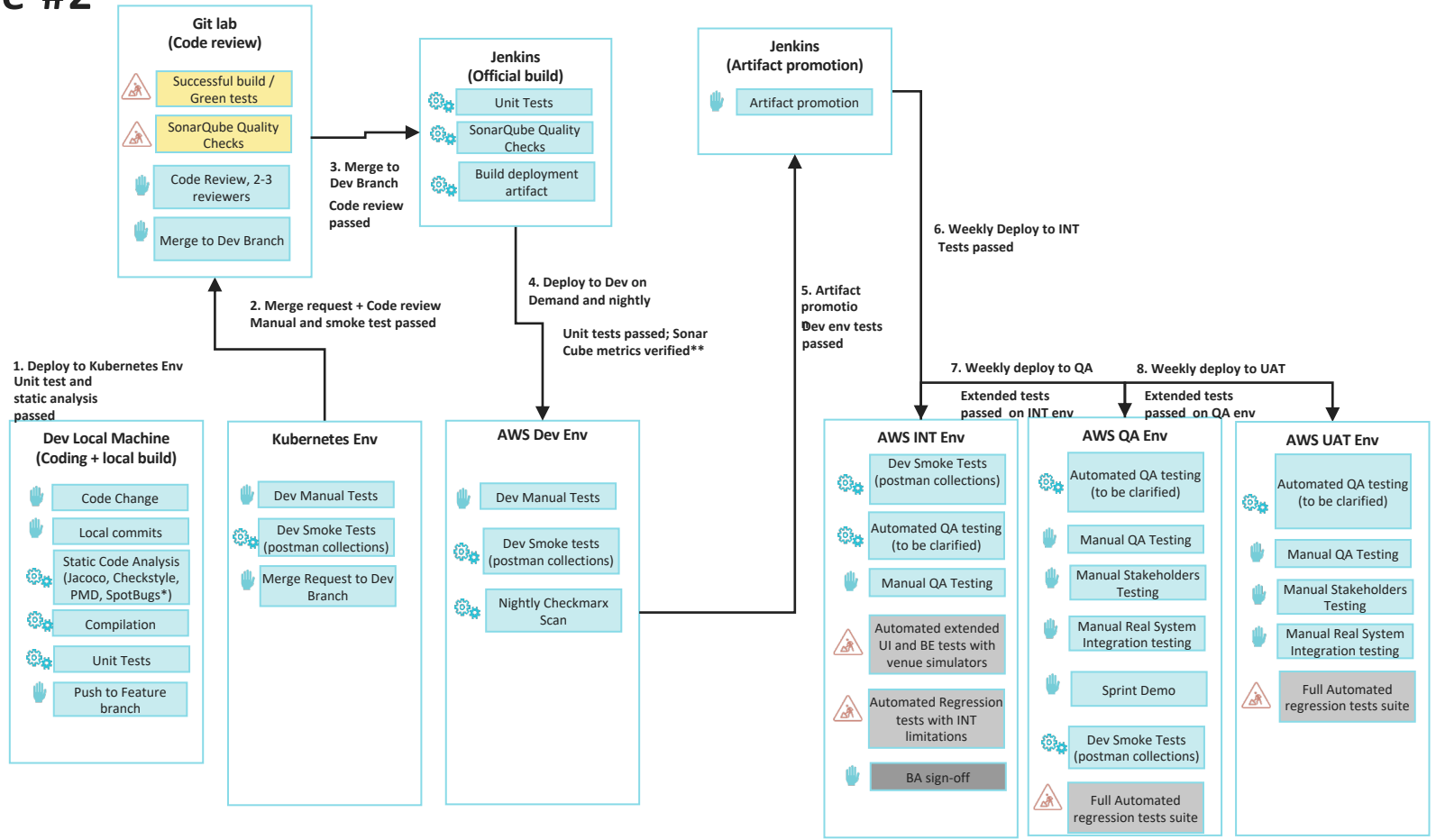
Value Stream Analysis



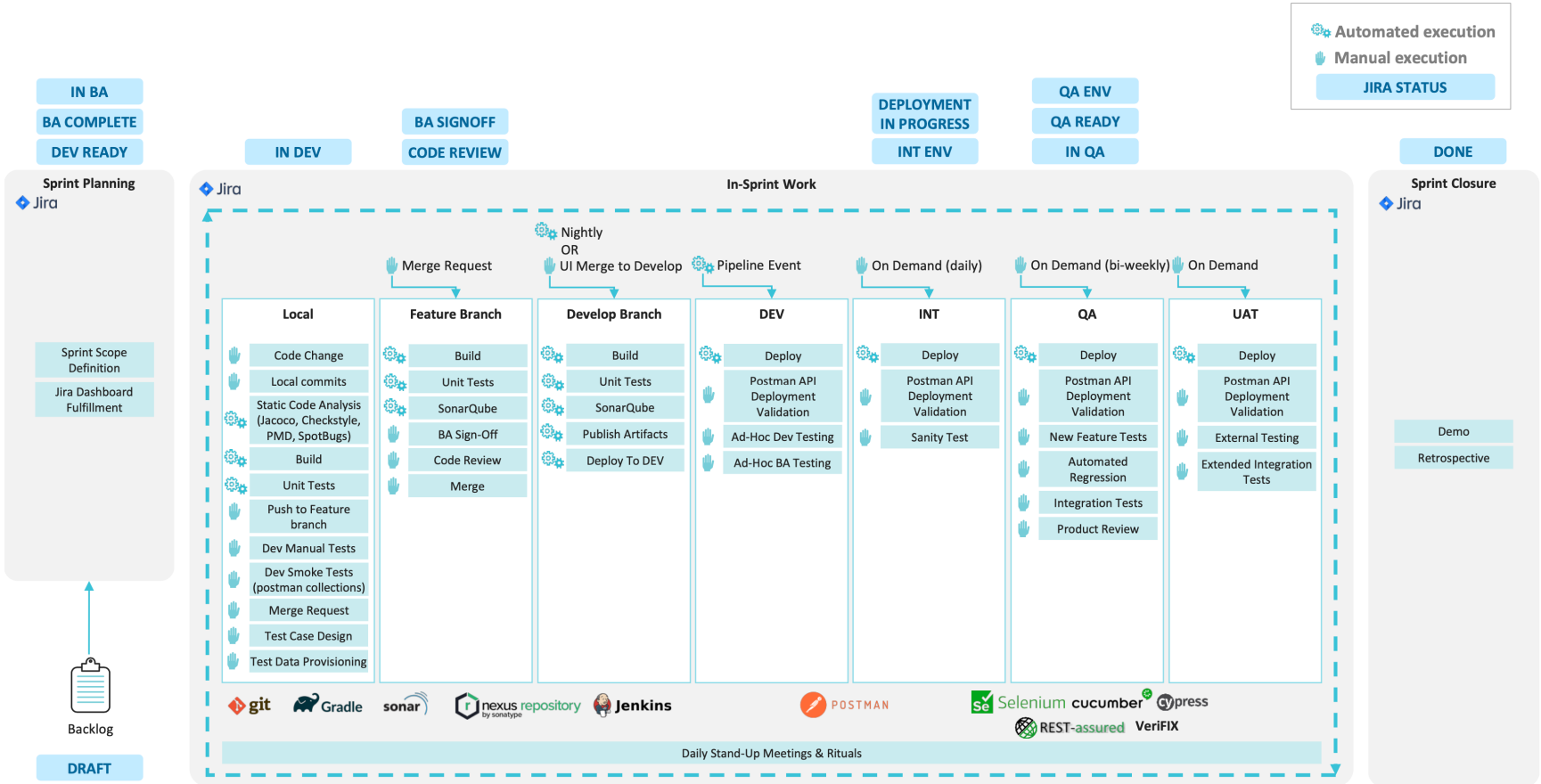
Example - Value Stream



Example #2

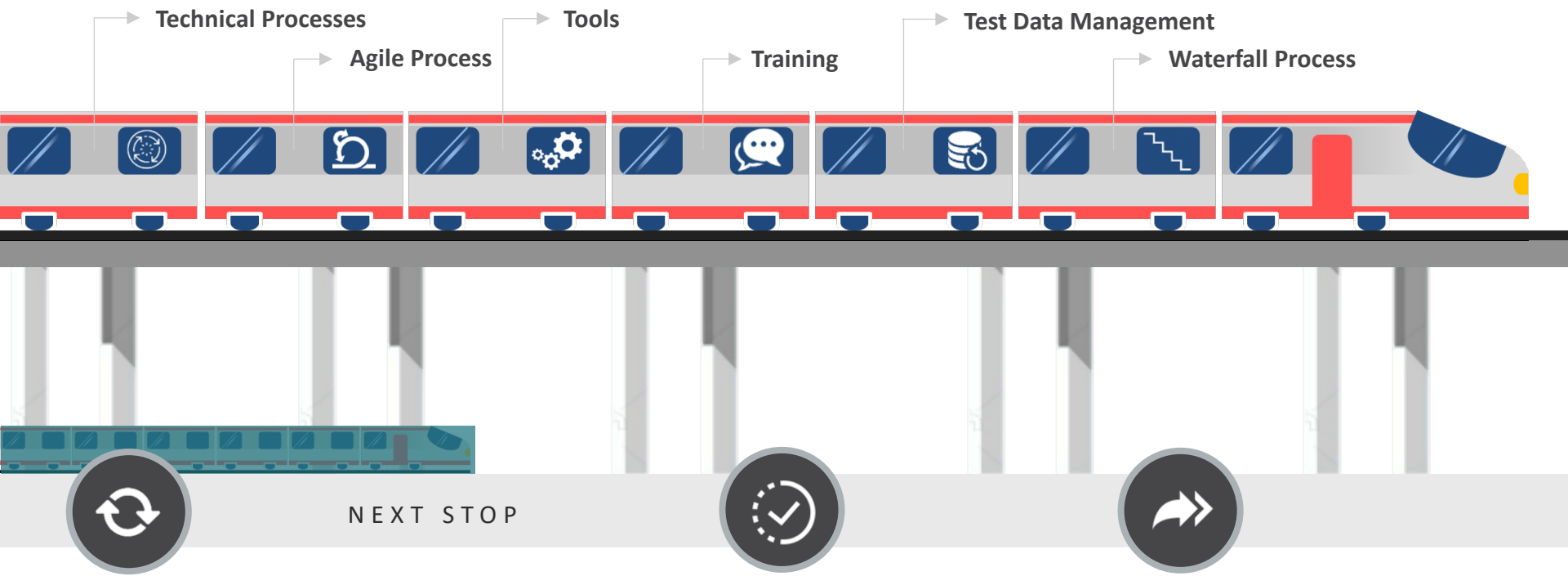

Git lab
(Code review)
Successful build / Green tests
SonarQube Quality Checks
Code Review, 2-3 reviewers
Merge to Dev Branch
3. Merge to Dev Branch
Code review passed
Jenkins
(Official build)
Unit Tests
SonarQube Quality Checks
Build deployment artifact
Jenkins
(Artifact promotion)
Artifact promotion
6. Weekly Deploy to INT
Tests passed
2. Merge request + Code review
Manual and smoke test passed
4. Deploy to Dev on Demand and nightly
Unit tests passed; Sonar Cube metrics verified**
5. Artifact promotio
Dev env tests passed
1. Deploy to Kubernetes Env
Unit test and static analysis passed
7. Weekly deploy to QA
8. Weekly deploy to UAT
Extended tests passed on INT env
Extended tests passed on QA env
Dev Local Machine
(Coding + local build)
Code Change
Local commits
Static Code Analysis (Jacoco, Checkstyle, PMD, SpotBugs*)
Compilation
Unit Tests
Push to Feature branch
Kubernetes Env
Dev Manual Tests
Dev Smoke Tests (postman collections)
Merge Request to Dev Branch
AWS Dev Env
Dev Manual Tests
Dev Smoke tests (postman collections)
Nightly Checkmarx Scan
AWS INT Env
Dev Smoke Tests (postman collections)
Automated QA testing (to be clarified)
Manual QA Testing
Automated extended UI and BE tests with venue simulators
Automated Regression tests with INT limitations
BA sign-off
AWS QA Env
Automated QA testing (to be clarified)
Manual QA Testing
Manual Stakeholders Testing
Manual Real System Integration testing
Sprint Demo
Dev Smoke Tests (postman collections)
Full Automated regression tests suite
AWS UAT Env
Automated QA testing (to be clarified)
Manual QA Testing
Manual Stakeholders Testing
Manual Real System Integration testing
Full Automated regression tests suite

Example #2 - Fixed

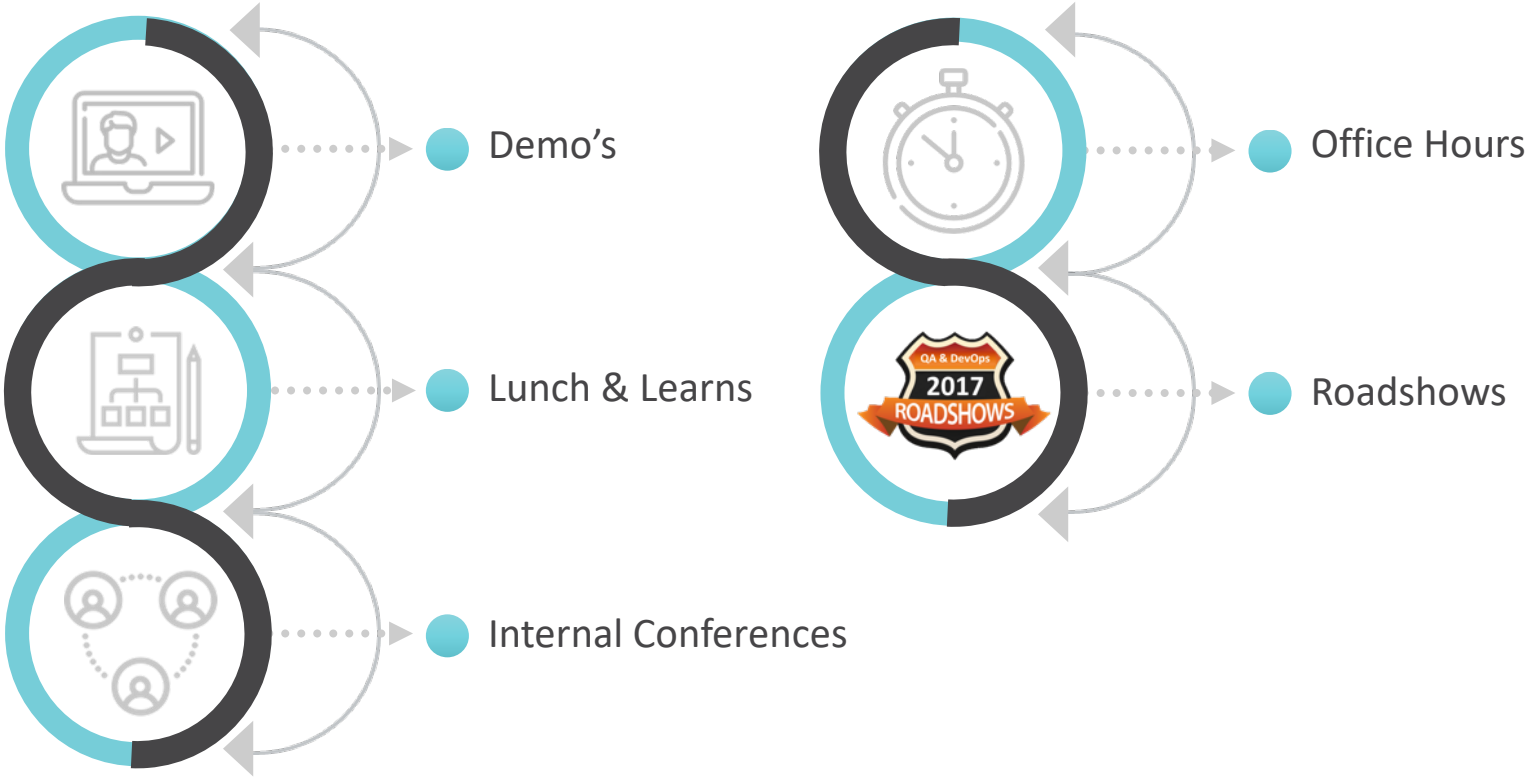


PEOPLE

Empower People to Drive Change



Build a Community



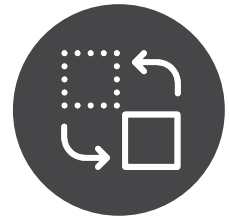
Common Challenges

Lack of **funding**

Project delivery **mindset**

3rd party **dependencies**

Questions on direction



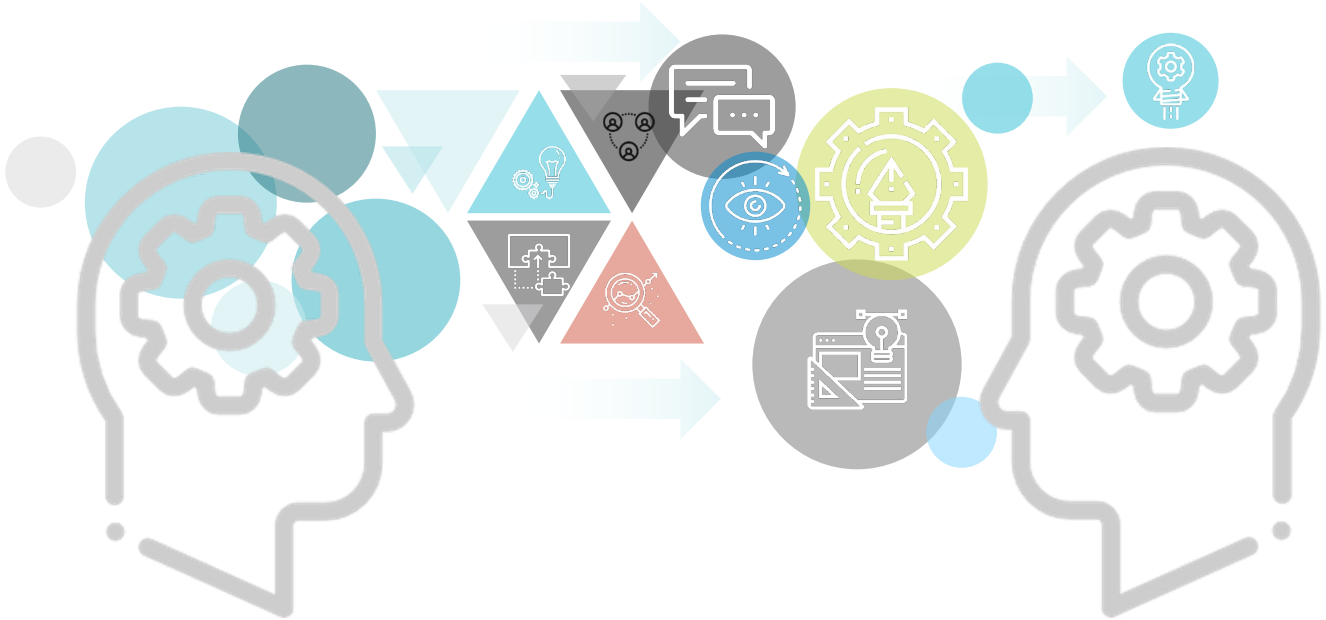
Lack of **Technical** resources

Top down and bottom up support is critical

BUGMAN31

Take Inventory of Your Team

Identify Your Future Leaders



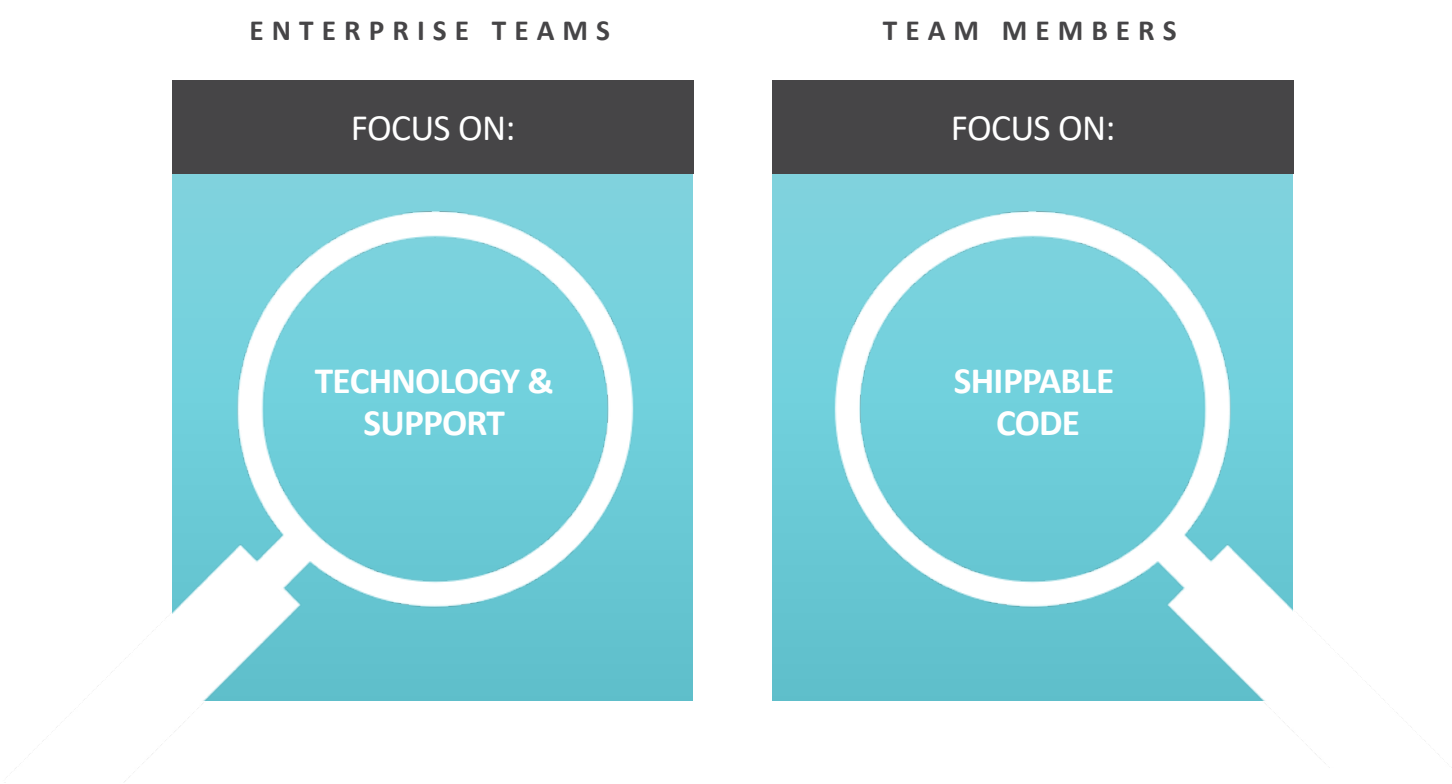
PEOPLE MANAGERS

TECHNOLOGISTS

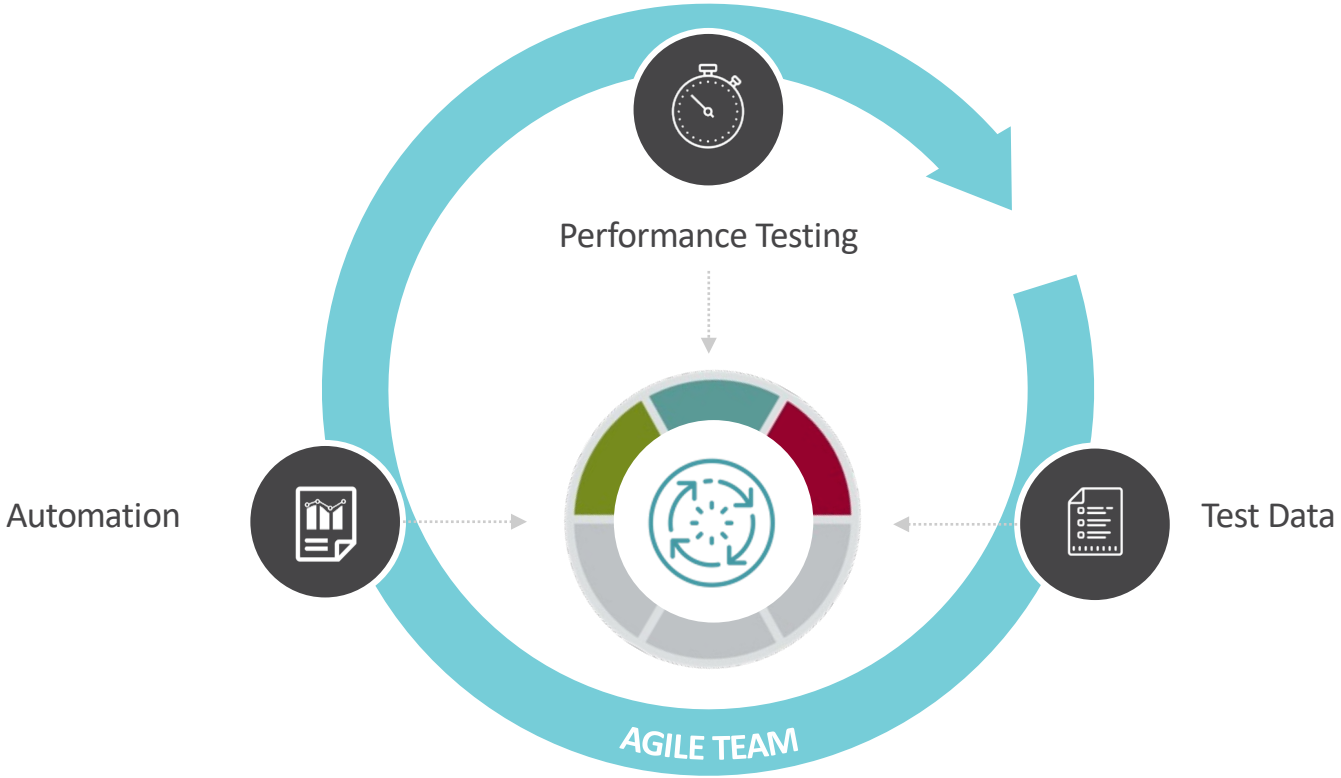
BUGMAN31

Use multiple levers to get more technical

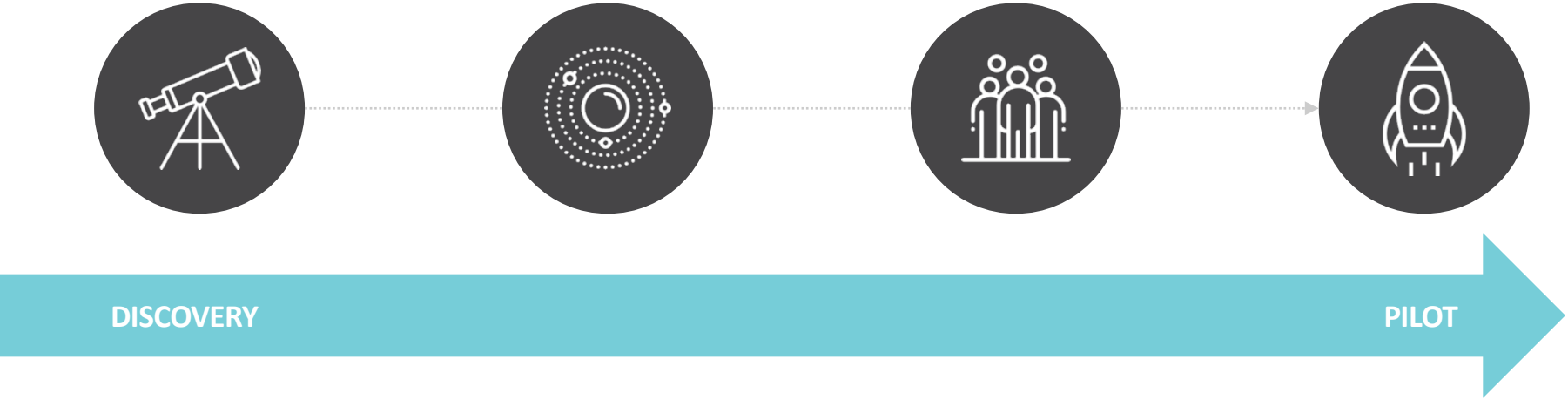
Enterprise Groups Focus on Enablement



Embrace DevOps for Testing



Pilot Like Crazy

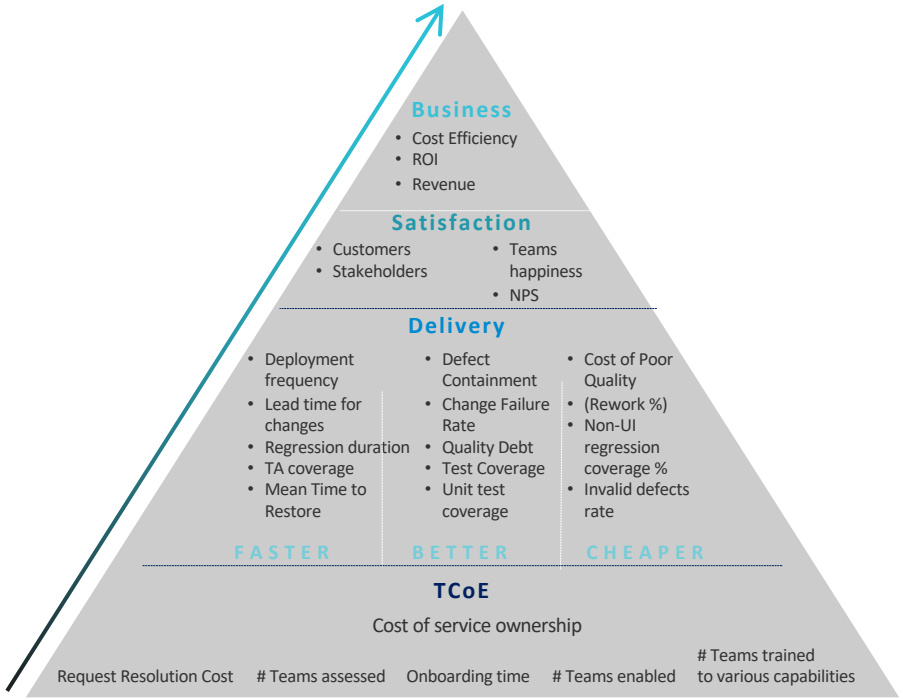


Metrics For Tracking Success

KPIs drive culture

Delivery pipelines (dynamic view)

Best practices (static view)



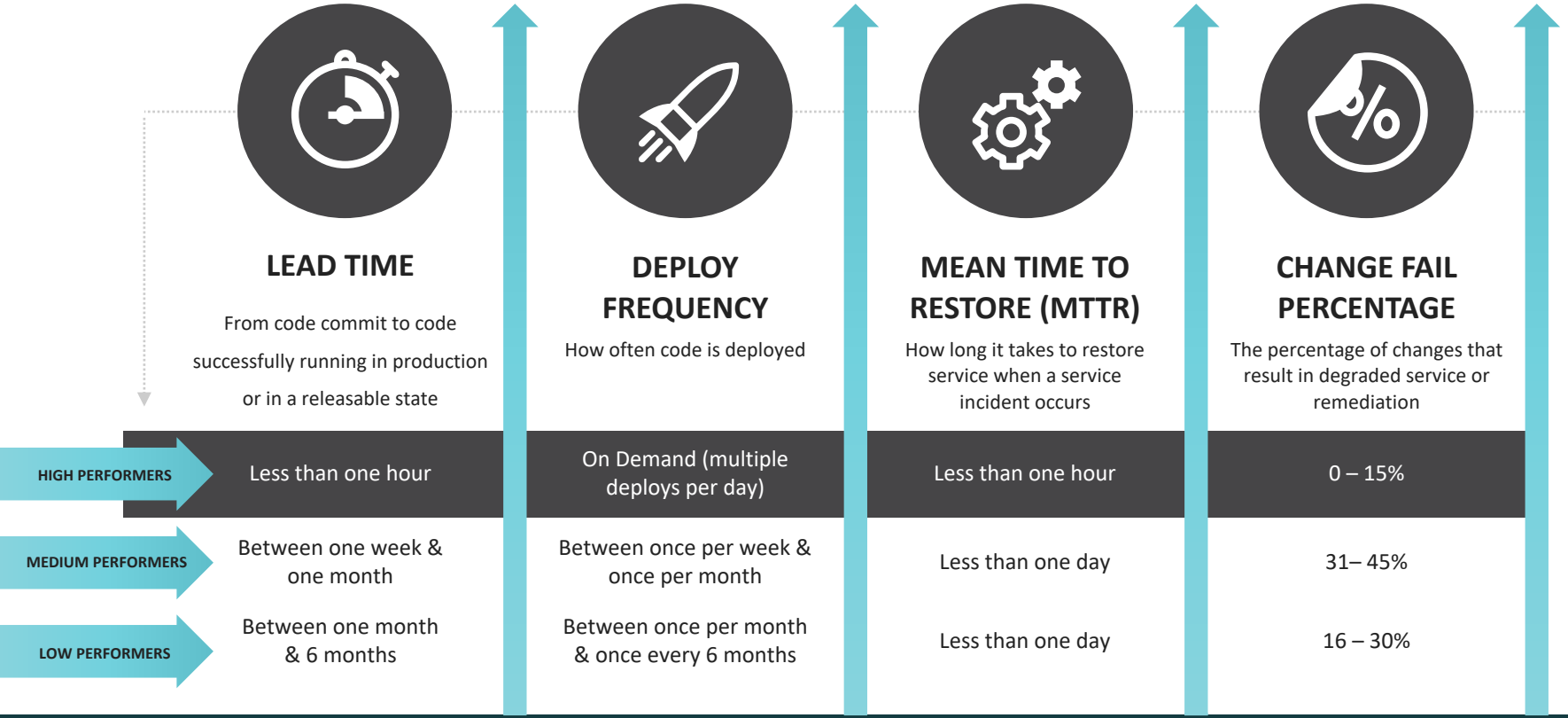
Business KPIs

Value & Satisfaction KPIs

Delivery pipeline: flow KPIs

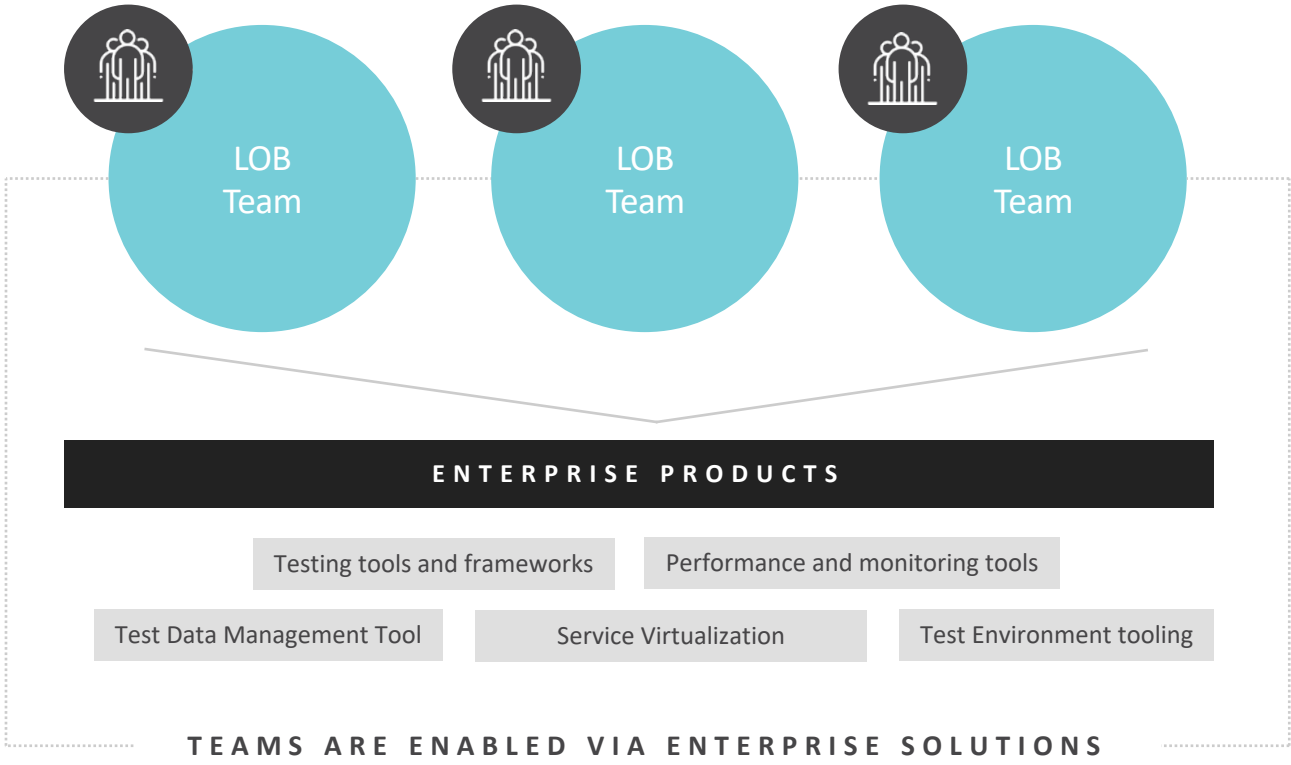
Practices KPIs

Proven Metrics



BUGMAN31

Continuous Testing Results I've Experienced



Lessons Learned for Successful Transformation

Potential Issues	Common challenges
Leadership Support	• Leadership support is not clearly articulated
Right organization structure	• Teams are fully driven by their vertical's leadership and transformation priorities have close to zero weightage • Transformation Stakeholder is not able to drive changes inside verticals in scope of transformation
Alignment on goals	• With no metrics available, current state is perceived as good enough, thus no motivation to change • Middle managers are not involved, transformation is viewed as the sole responsibility of consultants / Transformation Manager
Change across all specializations	• Testing transformation requires changes to upstream processes • Effective test automation requires changes to development and devops practices
Buy-in from the teams	• Teams are not motivated, but enforced only • Teams perceive the change as unnecessary and time devouring • Business need and Destination state is not clear for the teams • Lack of drivers on the team level
Plan to supplement missing skills	• Teams may be lack knowledge of testing best practices
Resources for the change	• Harder to allocate time from the team to implement technical improvements, rather than process

How to DevOps your Testing Strategy

An Exercise in Value Stream Analysis

Adam Auerbach