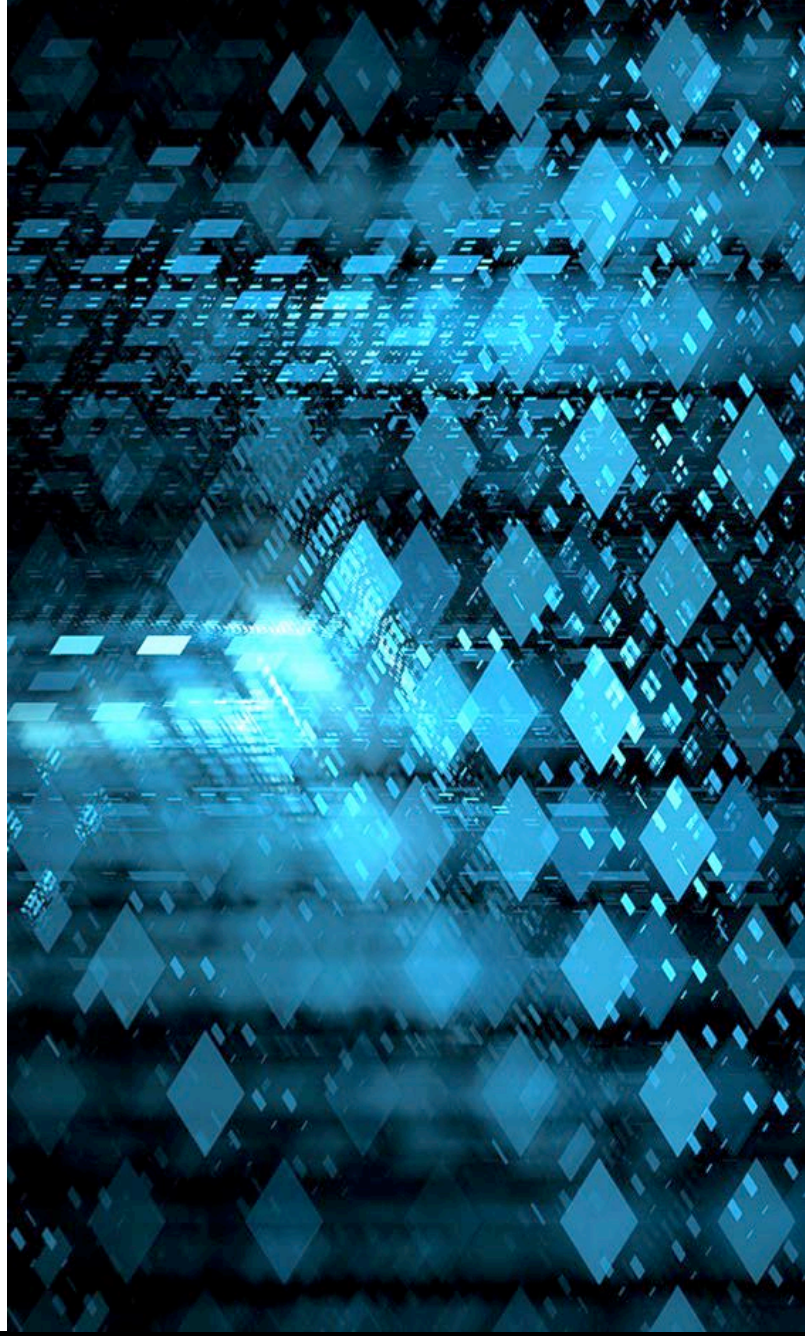


Can Agile Methodology Survive without DevOps Techniques?

Hasan Yasar

Technical Manager

Software Engineering Institute - CMU

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM17-0685

@securelifecycle

Topics

- *Background*

  Agile Manifesto + Methodologies

  DevOps: Reduce time to market

  Barriers on Agile methodologies

  How DevOps will overcome?

  With or Without Agile?

  Conclusion

@securelifecycle

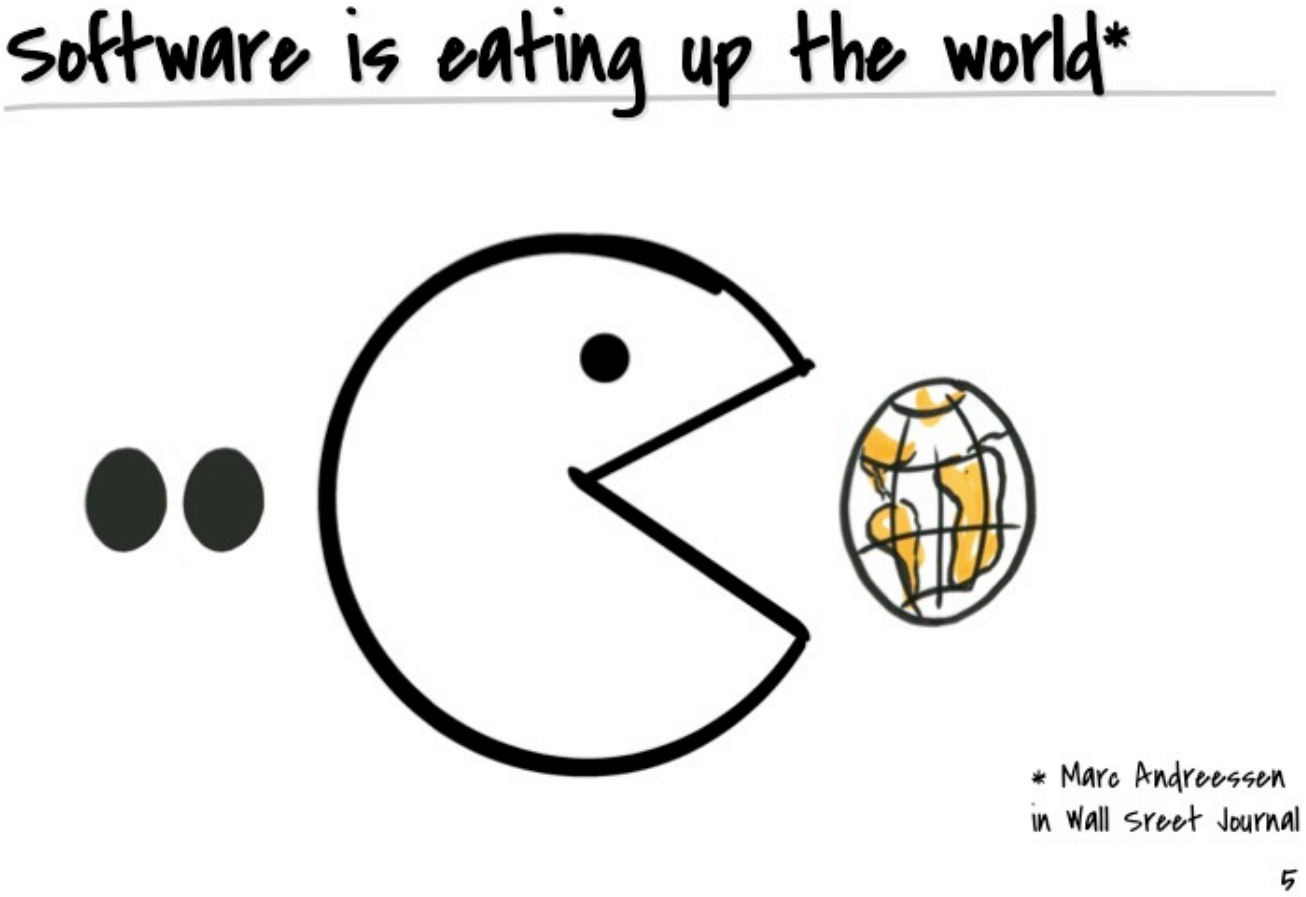
Background

- The Software Engineering Institute (SEI) is a Federally Funded Research and Development Center (FFRDC)
- Source of CMMI, TSP, PSP
- Research software and cybersecurity problems of considerable complexity, create and test innovative technologies, and transition maturing solutions to widespread use.
- Assisted numerous government organizations in modernizing their software development practices in the spirit of DevOps principles.

- 25+ years of software development experiences
- Certified Scrum Practitioner
- Various roles through SDLC ; Manager, Architect, Tester, Developer, QA, IT Manager, Project Manager, VP…
- Started with waterfall in 1990
- Started with agile in 2003
- Started with DevOps in 2010
- Instructor on delivering DevOps course at CMU, SEI since 2015

@securelifecycle

The world we live in..



@securelifecycle

Common question…

How can I implemented DevOps process and techniques in my team/ directorate / project / organization /unit … ?

How to assess the current state?
Where are the productivity bottlenecks?
Whom to train on what?
What and how to measure?
How to monitor?

Lead to another question…

Can Agile Methodology Survive without DevOps Techniques?

@securelifecycle

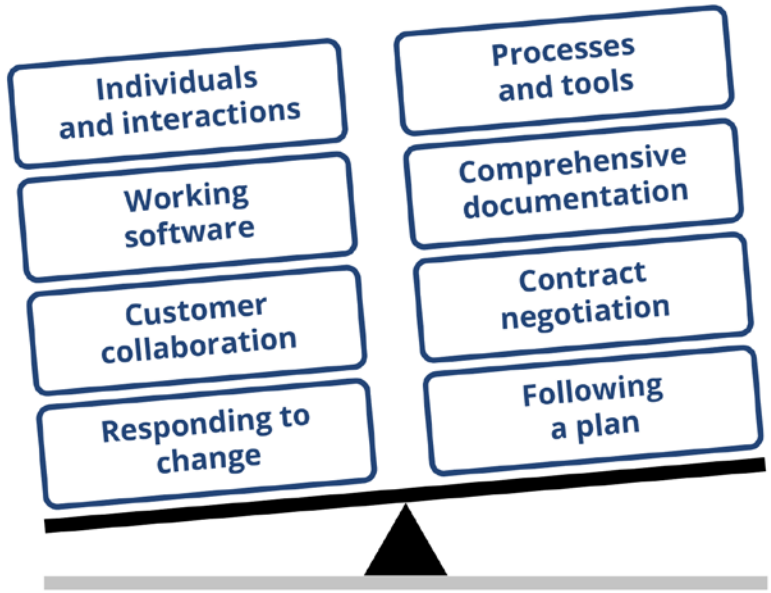
Topics

- Background
- Agile Manifesto + Methodologies
- DevOps: Reduce time to market
- Barriers on Agile methodologies
- How DevOps will overcome?
- With or Without Agile?
- Conclusion

@securelifecycle

Manifesto for Agile Software Development

Through this work we have come to value:



That is, while there is value in the items on the right, we value the items on the left more.

http://www.agilemanifesto.org/

@securelifecycle

Twelve Agile Principles Support the Manifesto

1. Highest priority is satisfy the customer through early and ***continuous delivery*** of software
2. Welcome ***changing requirements***, even late in development
3. Deliver working software ***frequently***, from a couple of weeks to a couple of months
4. ***Business people and developers*** must work together daily throughout the project
5. Build projects around motivated individuals. ***Provide environment and support*** they need
6. The most efficient and effective method of conveying information to and within a development team is ***face-to-face conversation***
7. ***Working software*** is the primary measure of progress
8. Agile processes promote sustainable development. The Sponsors, developers and users should be able to maintain a c***onstant pace indefinitely***
9. ***Continuous*** attention to technical excellence and good design enhances agility
10. ***Simplicity—the*** art of maximizing the amount of work not done—is essential
11. The best architectures, requirements, and designs emerge from ***self-organizing teams***
12. At regular intervals, the team reflects on how to become more effective, then ***tunes and adjusts*** its behavior accordingly.

Some Observable Characteristics of Agile Implementations

Iterative —elements are expected to move from skeletal to completely fleshed out over time, not all in one step

Incremental —delivery doesn't occur all at once

Collaborative —progress is expected to be made by stakeholders and the development team working collaboratively throughout the development timeframe

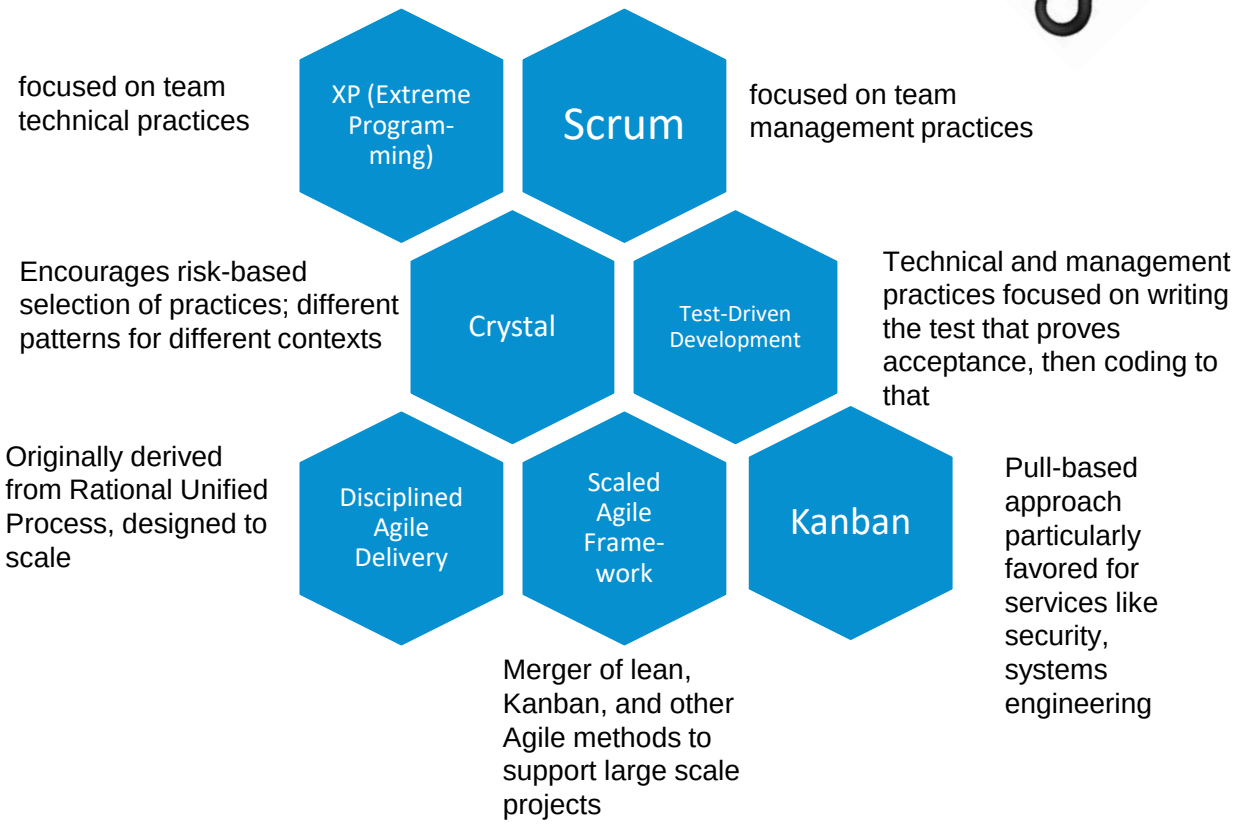
Loosely-coupled Architecture —multiple self-organizing, cross-functional teams work concurrently on multiple product elements (e.g., requirements, architecture, design, and the like) for multiple loosely coupled product components

Dedicated —team members are allowed to focus on the tasks within an iteration/release as opposed to multi-tasking across multiple projects

Time-boxed or *Flow-based* —relatively short-duration development cycles that permit changes in scope rather than changes in delivery time frame

Adapted from Nidiffer, Miller, & Carney. Potential Use of Agile Methods in Selected DoD Acquisitions: Requirements Development & Management, SEI-2013-TN-006

Many Methods Generally Termed

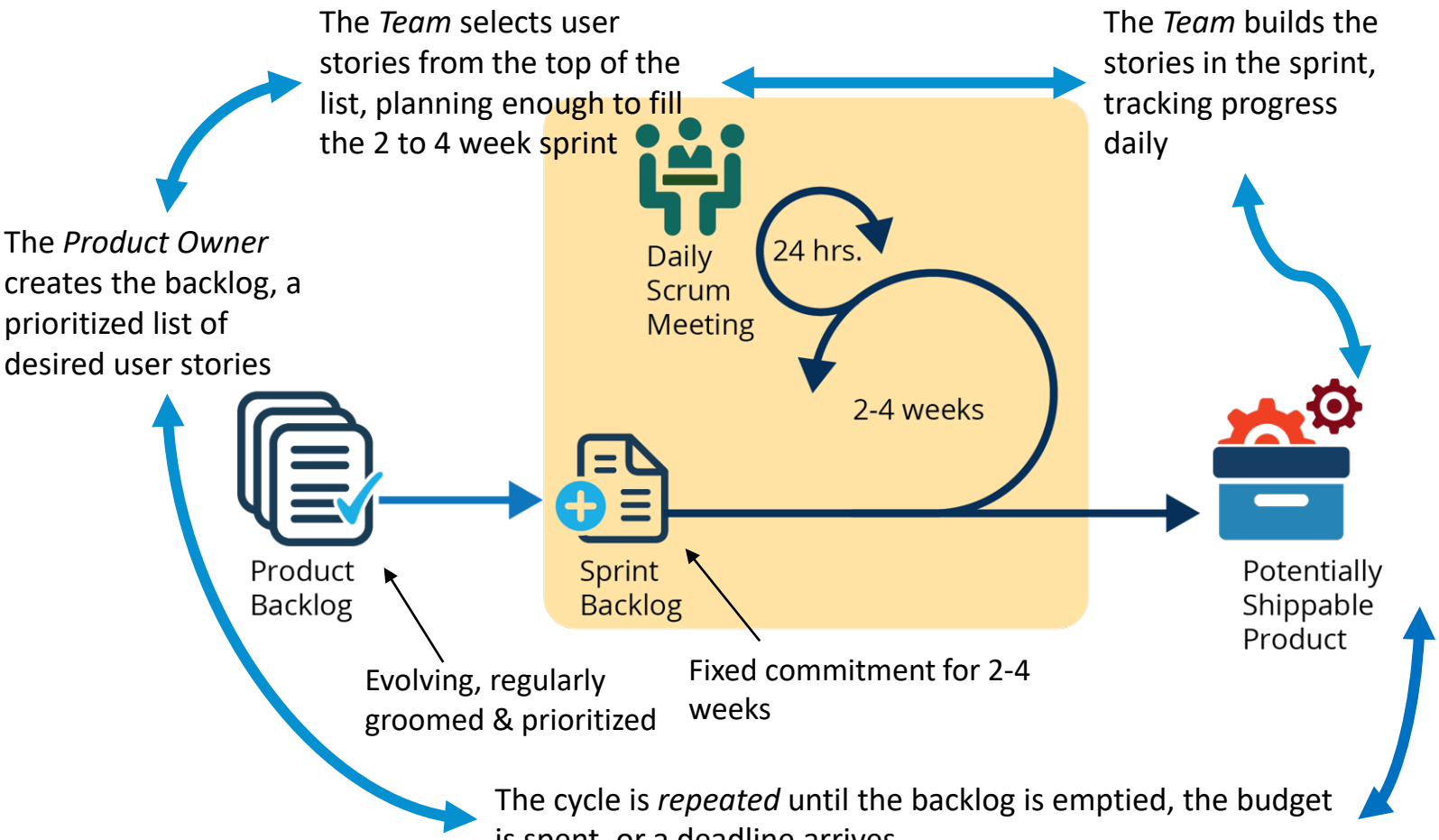


Common thought:
any method that is an alternative to documentation driven, heavyweight software development processes is probably an Agile method.

@securelifecycle

Agile at the Team Level - Scrum

Scrum is an iterative, incremental methodology for managing agile software projects.



@securelifecycle

Agile at the Team Level - Kanban

Similar to Scrum in the sense that you focus on features as opposed to groups of features

select, plan, develop, test and deploy one feature before the next feature.

Principles:

- Visualize workflow,
- Limit WIP, under the “in progress”
- Pull work form column to column,
- Measure Lead Time: Monitor, adapt, improve

Measures and Progress are common in Agile Methods

Working software – the potentially releasable features created during each sprint.

Burn-down chart – tracks the work completed and work remaining.

- two views – the current sprint and the overall project
- includes the hours of effort remaining and the number of tasks remaining in for the sprint or project.

Velocity – measures the number of user stories completed in a sprint and can be used to predict the completion date range.

Cyclomatic complexity – a measure of the "goodness" of the software that is used to establish the need for refactoring.

Topics

- Background
- Agile Manifesto + Methodologies
- ▶ *DevOps: Reduce time to market*
- Barriers on Agile methodologies
- How DevOps will overcome?
- Set the expectations
- Final Thoughts

@securelifecycle

What is DevOps?

DevOps (a portmanteau of "development" and "operations") emphasizes communication, collaboration, and integration between software developers and information technology (IT) operations personnel. [1]

The history of DevOps

- Patrick Debois “Agile infrastructure and operations: how infra-gile are you?”, Agile 2008 Conference
- John Allspaw “ 10+Deploys per Day: Dev and Ops Cooperation”, Velocity 2009
- DevOpsDays, October 30th 2009, #DevOps term born

[1] http://en.wikipedia.org/wiki/DevOps

@securelifecycle

Who are Dev?

- Follow Agile methodologies
  - Using Scrum, Kanban and modern development approaches
  - Self directing, self managed, self organized
- Using any new technology
  - Each Dev has own development strategy
  - OpenSource,
- Allowed to have
  - Close relationships with the business
  - Software driven economy

Want to deliver software faster with new requirements…

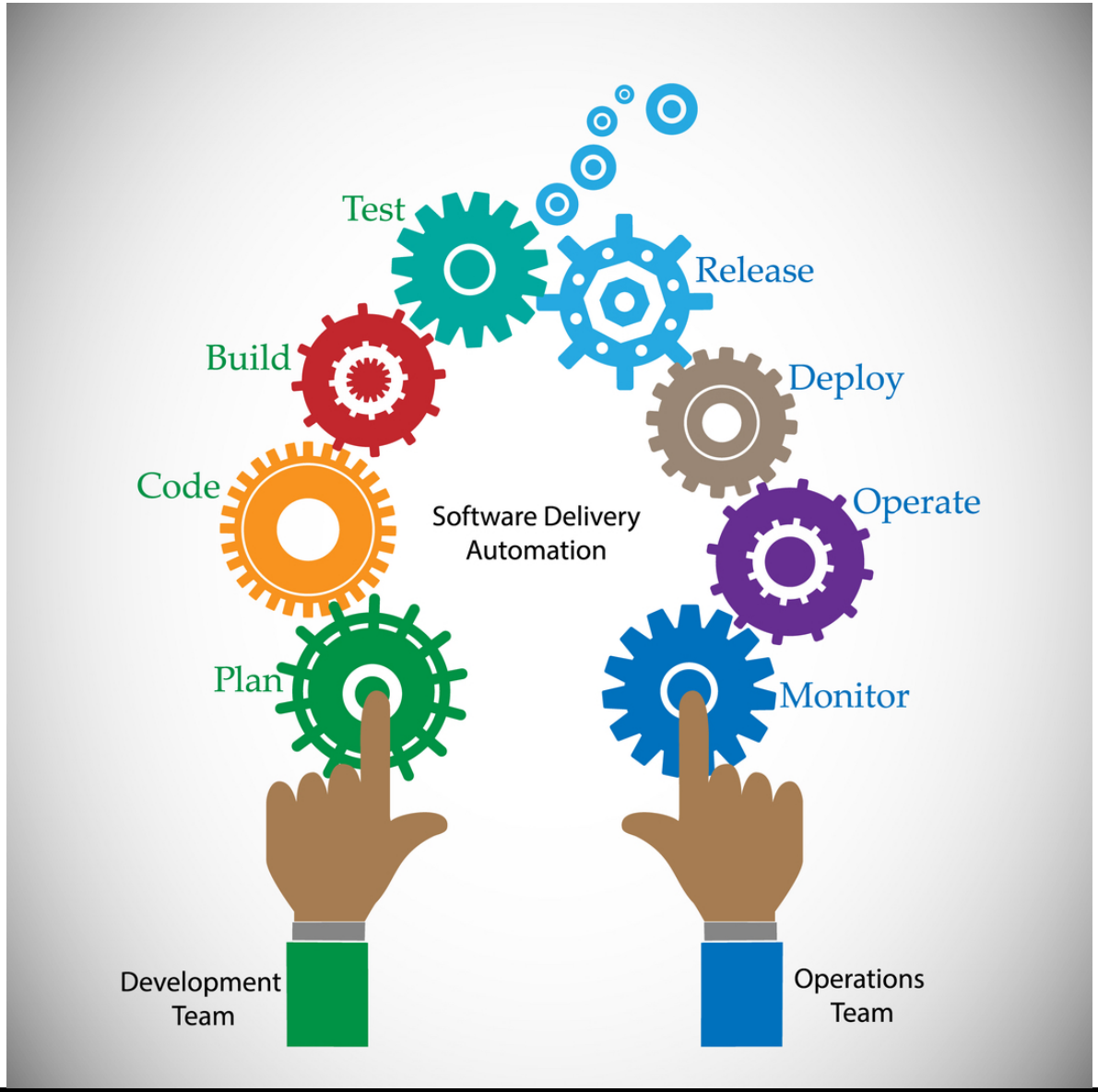
@securelifecycle

Who are Ops?

- Operations
  - Runs the application
  - Manages the infrastructure
  - Support the applications
- Operations provides (ITIL Says)
  - Service Strategy
  - Service Design
  - Service Transition
  - Service Operations

@securelifecycle

DevOps connects Dev to Ops to Dev



@securelifecycle

DevOps aims to Increase…

…the pace of **innovation**

…**responsiveness** to business needs

…**collaboration**

…software **stability and quality**

... continuous feedback

DevOps has four Fundamental Principles

Collaboration: between project team roles

Infrastructure as Code: all assets are versioned, scripted, and shared where possible

Automation: deployment, testing, provisioning, any manual or human-error-prone process

Monitoring: any metric in the development or operational spaces that can inform priorities, direction, and policy

DevOps promotes Collaboration

Heavy collaboration between Dev and Ops on:

- Design / Architecture decisions
- Environment / Network configuration
- Deployment planning
- Code Review

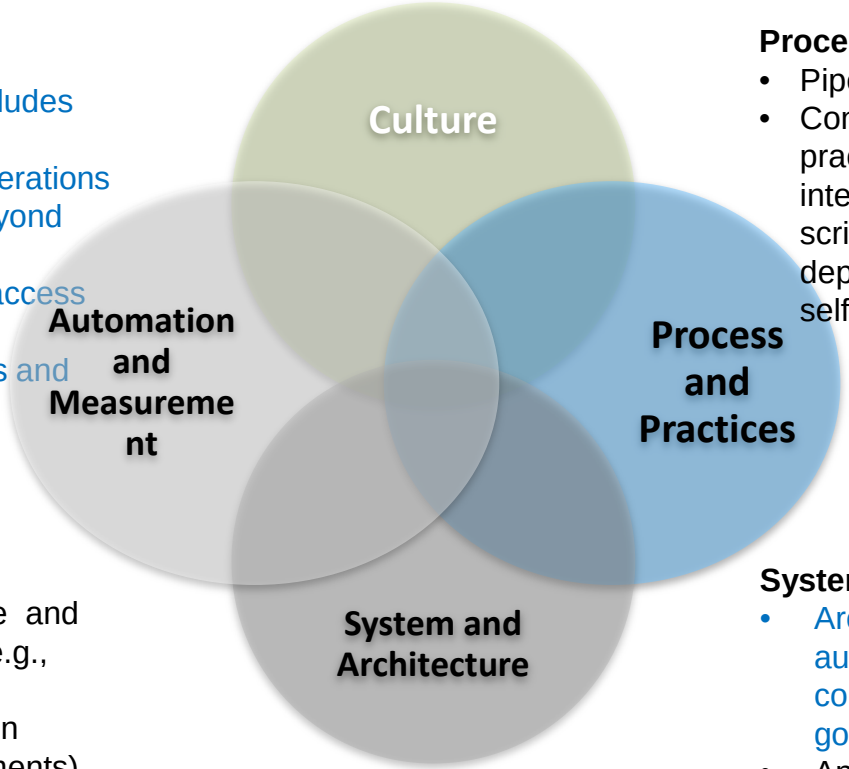
Constantly available open communication channels:

- Dev and Ops together in all project meetings
- Chat/Email/Wiki services available to all team members
- Dev / Ops report together as one project team

@securelifecycle

Multiple Dimensions of DevOps

Culture
- Developer and Ops collaborate (Ops includes security)
- *Developers* and Operations support releases beyond deployment
- Dev and Ops have access to stakeholders who understand business and mission goals



Process and Practices
- Pipeline streamlining
- Continuous-delivery practices (e.g., continuous integration; test automation; script-driven, automated deployment; virtualized, self-service environments)

Automation/ Measurement
- Automate repetitive and error-prone tasks (e.g., build, testing, and deployment maintain consistent environments)
- Static analysis automation (architecture health)
- Performance dashboards

System and Architecture
- Architected to support test automation and continuous-integration goals
- Applications that support changes without release (e.g., late binding)
- Scalable, secure, reliable, etc.

@securelifecycle

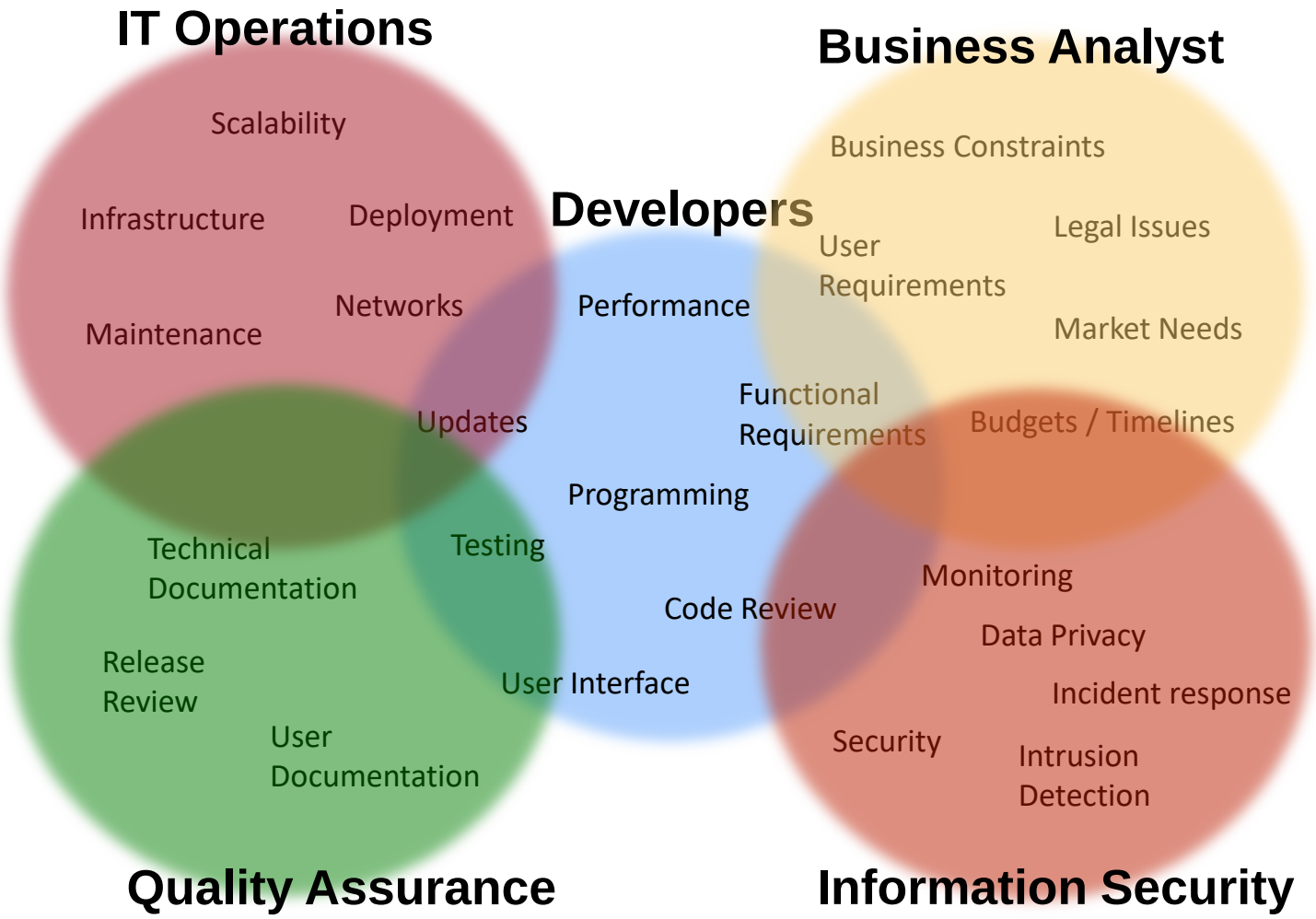
Topics

- Background
- Agile Manifesto + Methodologies
- DevOps: Reduce time to market
- *Barriers on Agile methodologies*
- How DevOps will overcome?
- With or Without Agile?
- Conclusion

Stakeholders Involvement

@securelifecycle

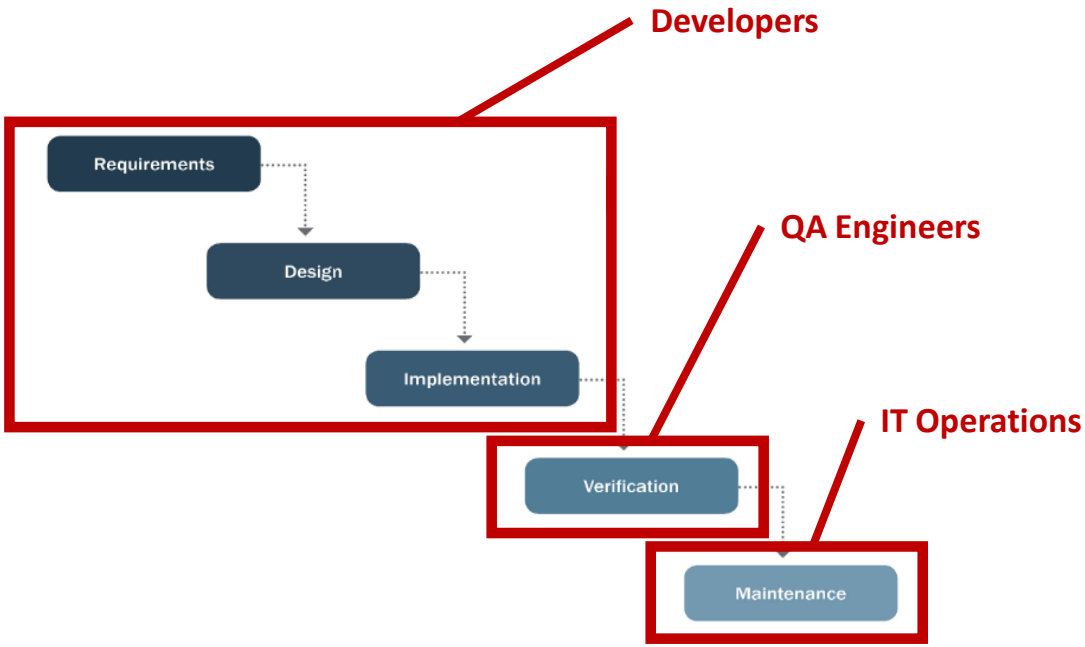
Teams that are too large to get everybody involved



@securelifecycle

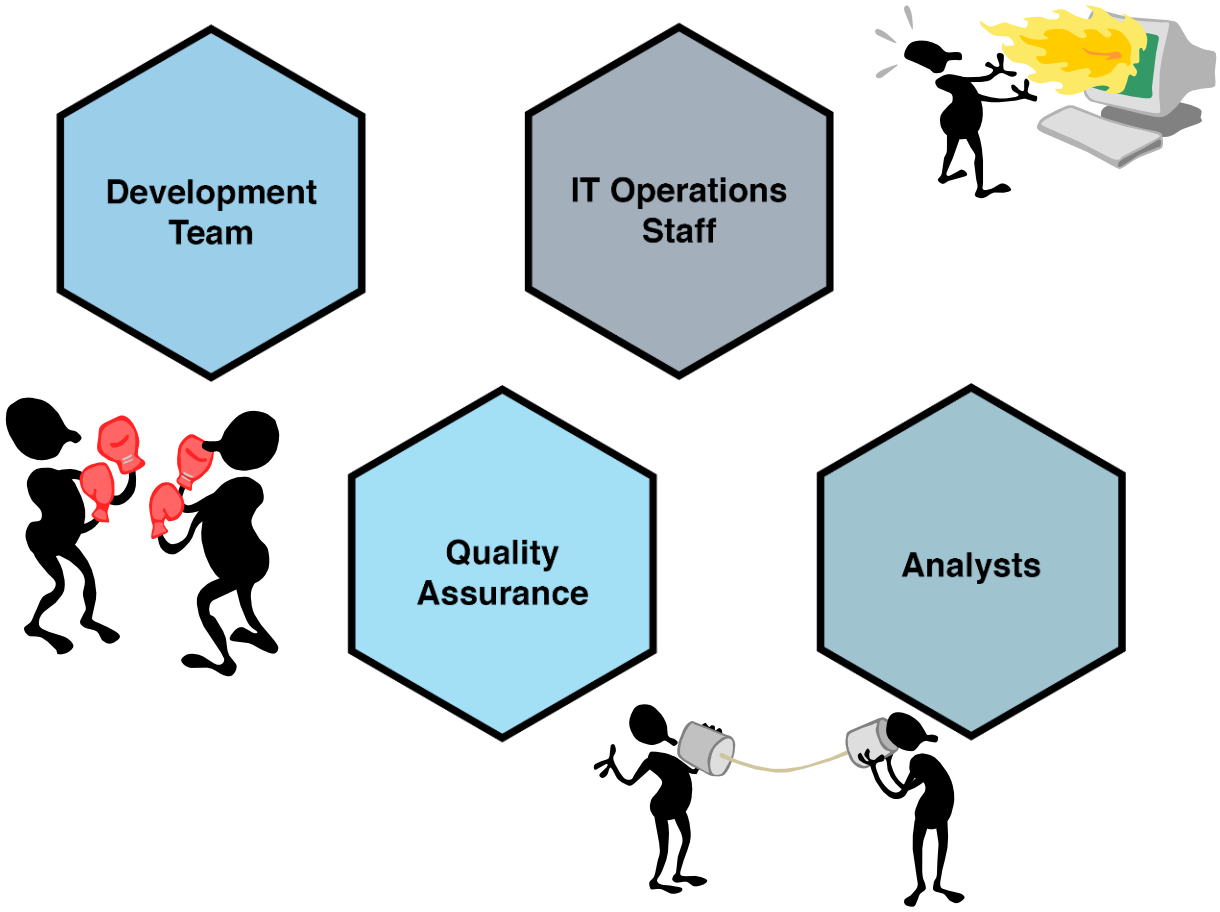
Silos Reinforce Waterfall

Teams have moved to Agile methodologies, but roles still align with waterfall methods



@securelifecycle

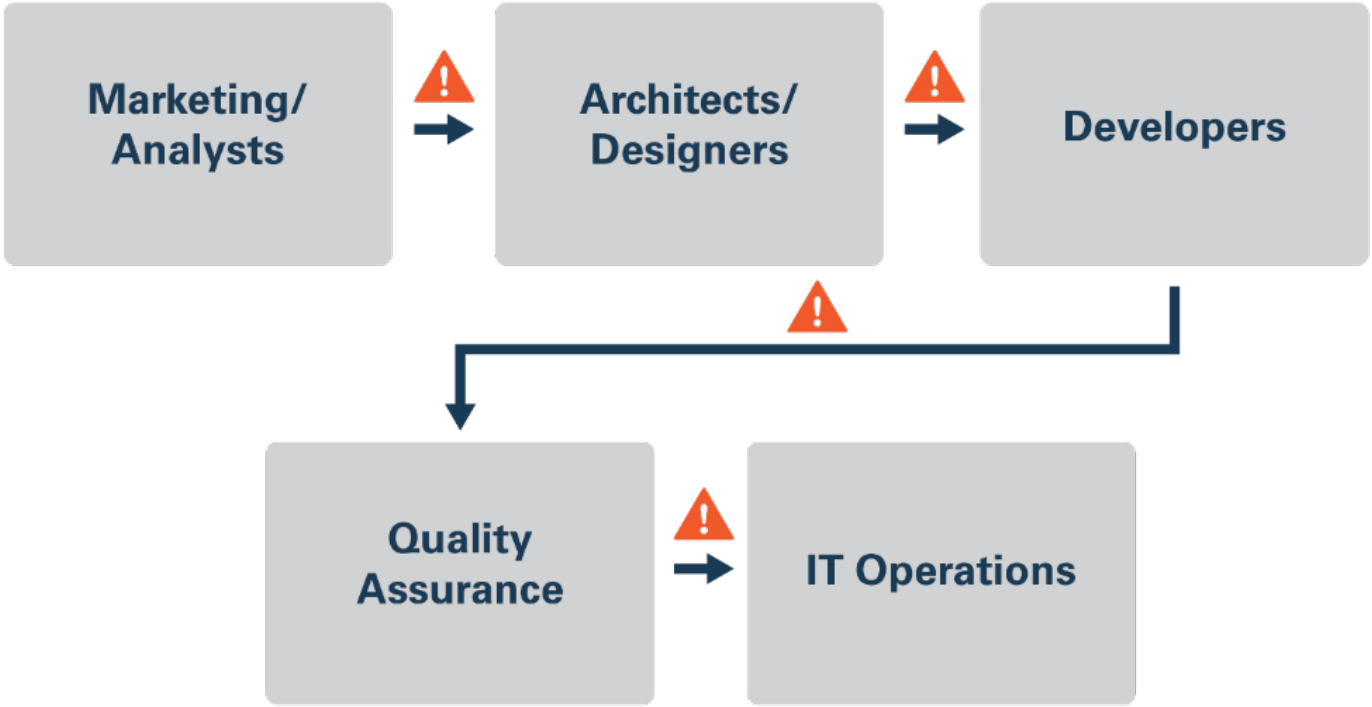
Silos Inhibit Collaboration and poor communication



@securelifecycle

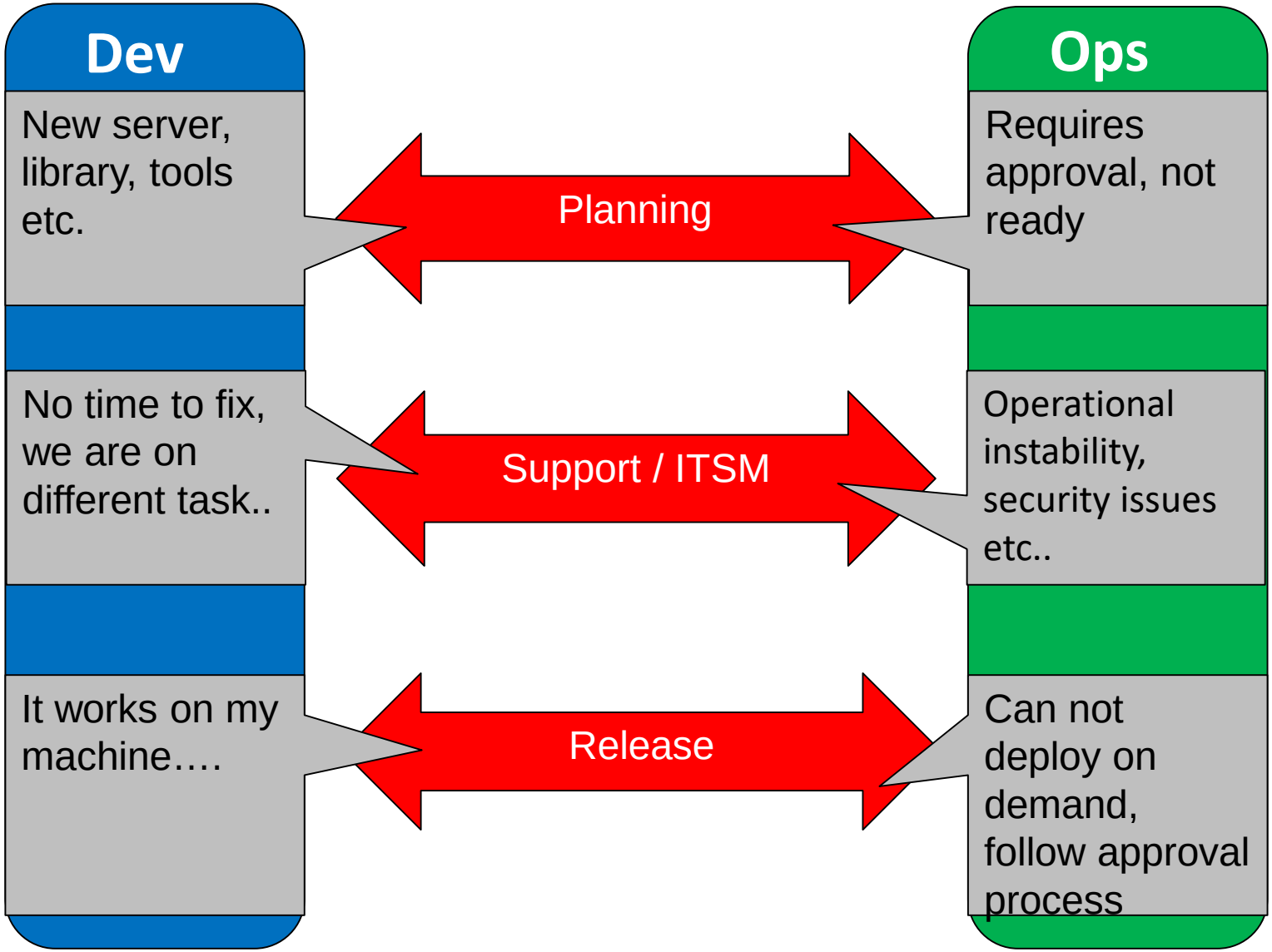
Artifact transitions

Every Transition of the System is a Risk

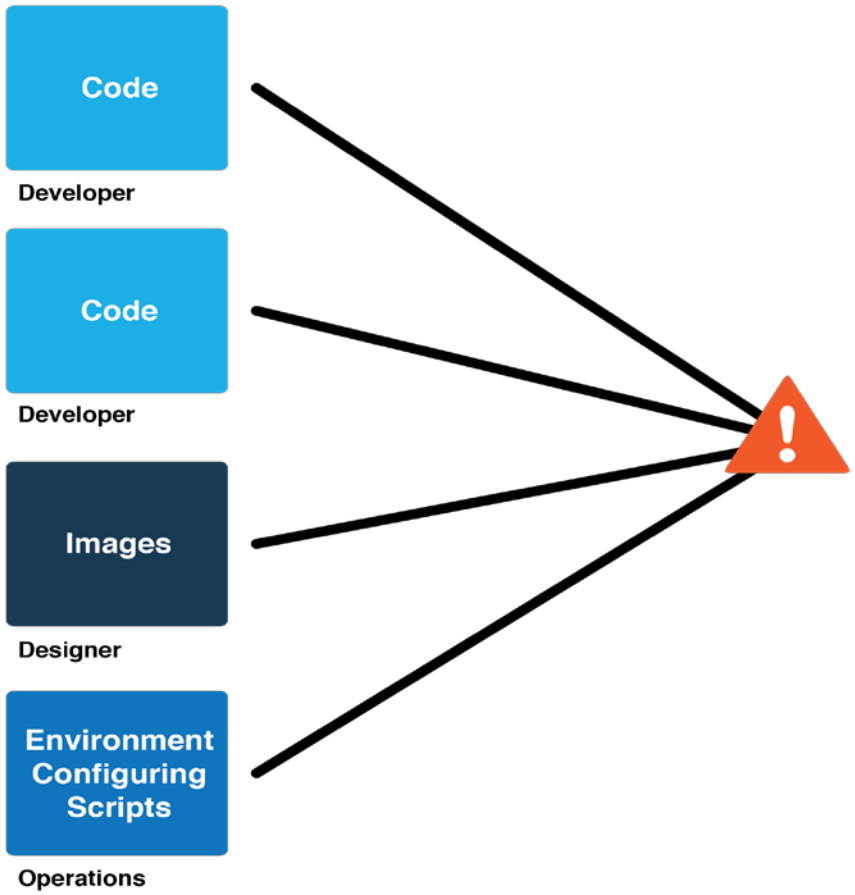


@securelifecycle

Constant conflict between Dev & Ops and also beyond

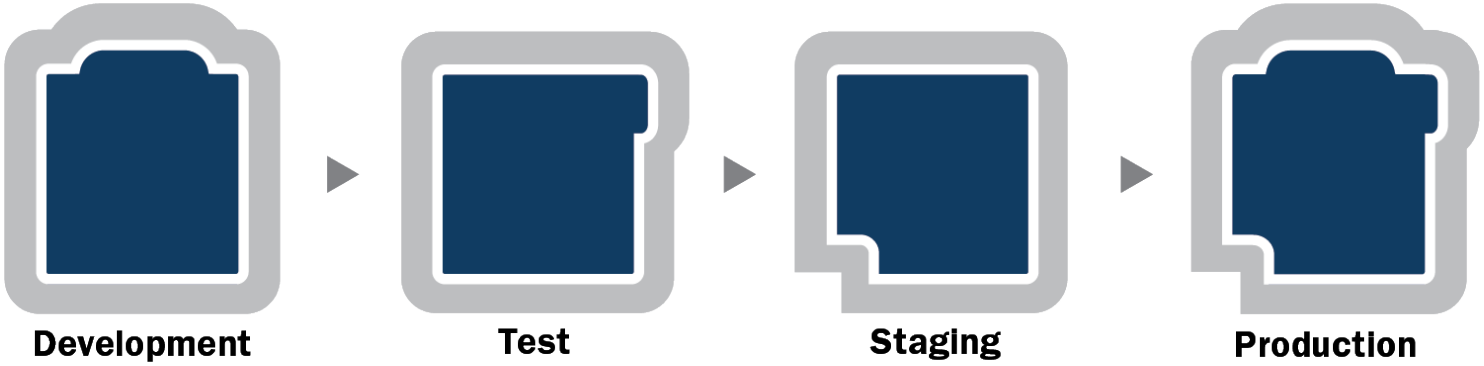


Late testing (integration)



- Integration hell,
- Lack of automation testing,
- Code freeze to integrate and test

Divergence between environments



Environments are independent, volatile, and easily manipulated.
Without care, they will diverge.

@securelifecycle

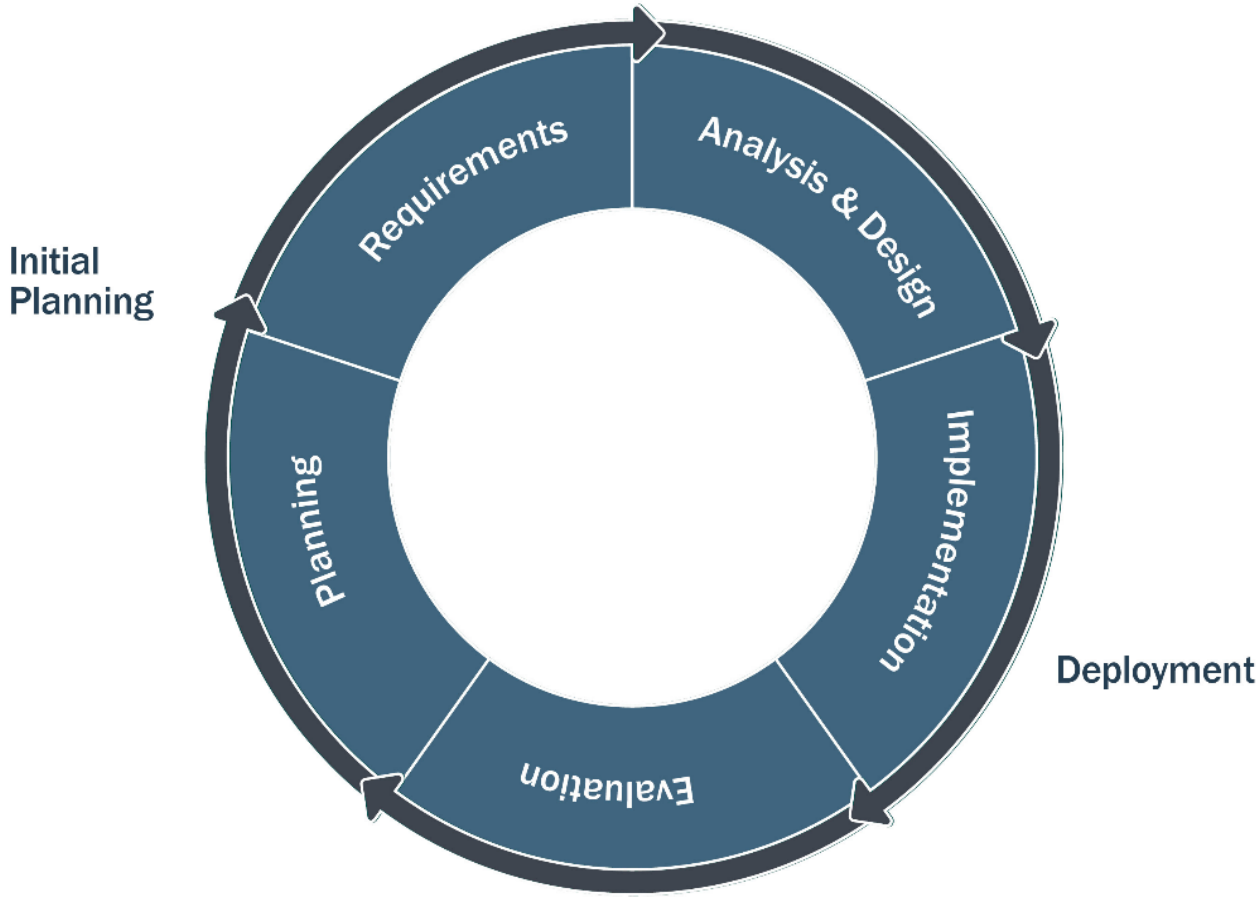
Lack of monitoring and measuring inappropriately

You will get what you measure if you do not monitor constantly

- What to monitor?
- Performance, Load/Tasking.
- Manual and based on instinct? Or Data
- Linked to business goals?
- Measuring of task? Or impact?
- Quality, Security, Reliability?

@securelifecycle

Agile Methods on Dev team

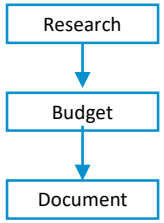


@securelifecycle

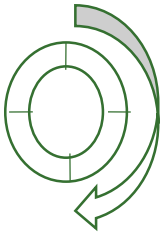
That results Agile development to …

Water - Scrum - Fall

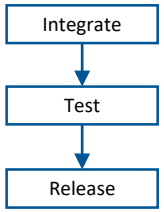
Business



Development



QA
Operations



Jez Humble, https://youtu.be/L1w2_AY82WY
Dave West, http://sdtimes.com/analyst-watch-water-scrum-fall-is-the-reality-of-agile/

@securelifecycle

Topics

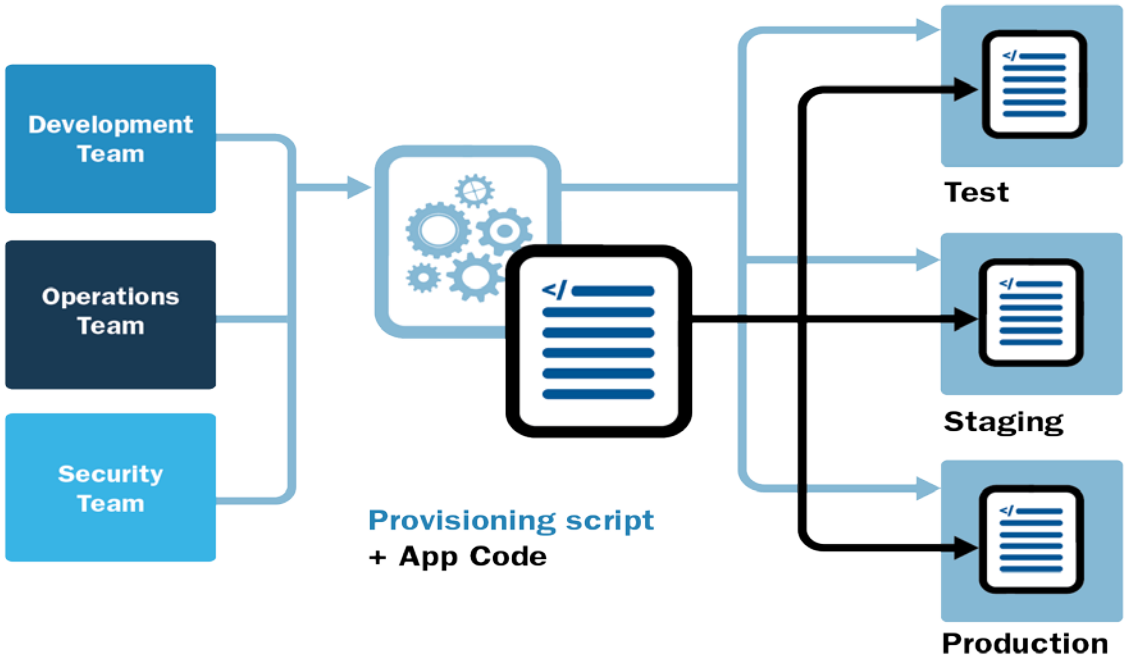
- Background
- Agile Manifesto + Methodologies
- DevOps: Reduce time to market
- Barriers on Agile methodologies
- *How DevOps will overcome?*
- Set the expectations
- Final Thoughts

@securelifecycle

Setup environment Parity

IaC (Infrastructure as Code):

A program that creates infrastructure



A concretely defined description of the environment is good material for conversation between team members.

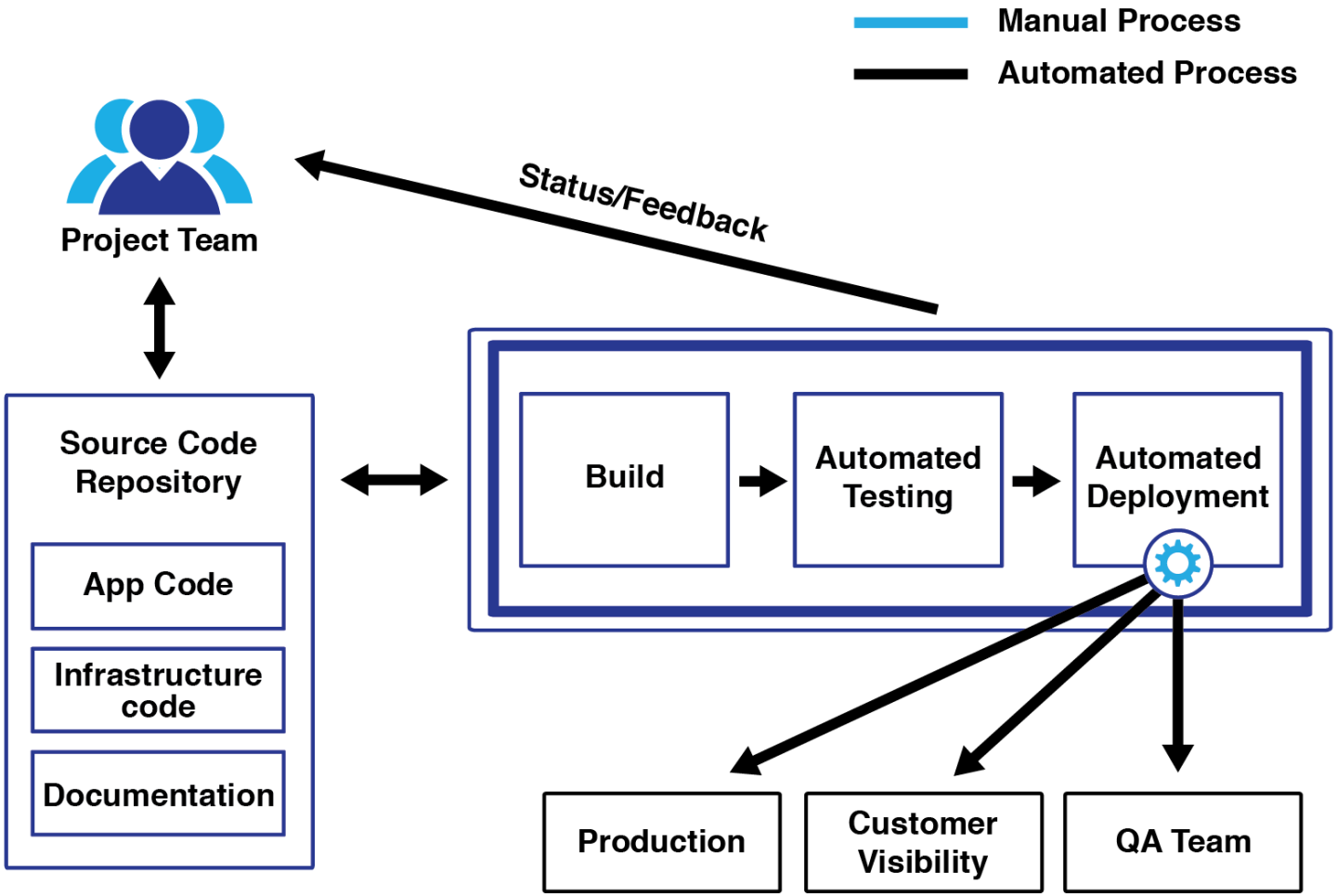
@securelifecycle

IaC Promotes Quality Attributes

- The process that creates and configures the infrastructure for your application is, itself, an application.

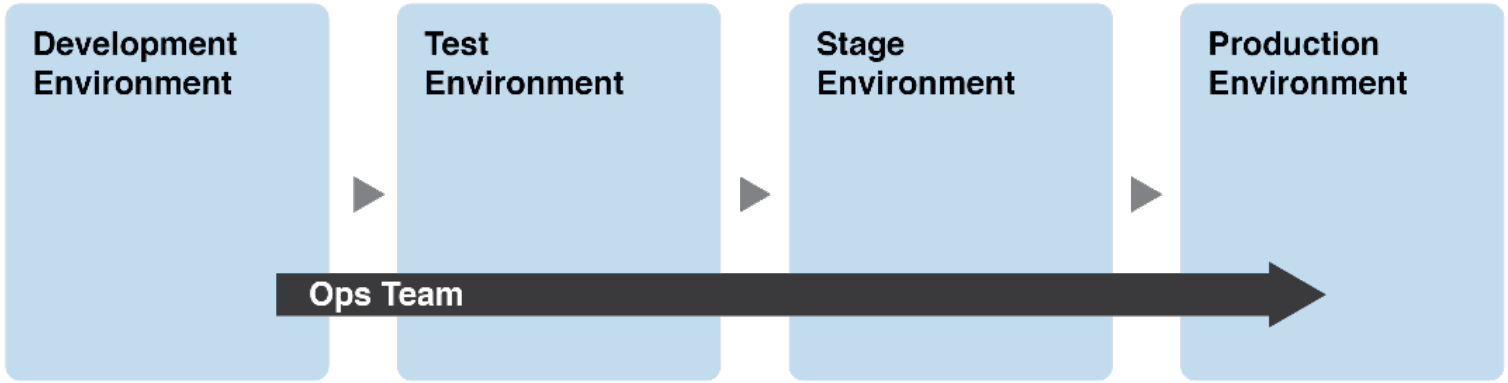
- As an application, environment creation and configuration is now:
  - Automatable
  - Repeatable
  - Versionable
  - Reviewable
  - Diffable
  - Testable (it works)
  - Human-readable
  - Verifiable (it is right)

Establish a Continuous Integration (CI)



@securelifecycle

Establish a Continuous Delivery /Deployment (CD)



Shift Left Operational Concerns Enforced by Continuous Delivery

@securelifecycle

DevOps with Agile : *People*

Heavy collaboration between all stakeholders

- Continuous Design / Architecture decisions
- Agreed on Environment / Network configuration
- Continuous Deployment planning
- Continuous Code Review

Constantly available open communication channels:

- Dev and Ops together in all project decision meeting
  - Virtually or physical but sharing common collaboration environment
- Chat/e-mail/Wiki services available to all team members

@securelifecycle

DevOps with Agile : *Process*

Establish a *process* to enable *people* to succeed using the *platform* to develop Secure application

Such that;

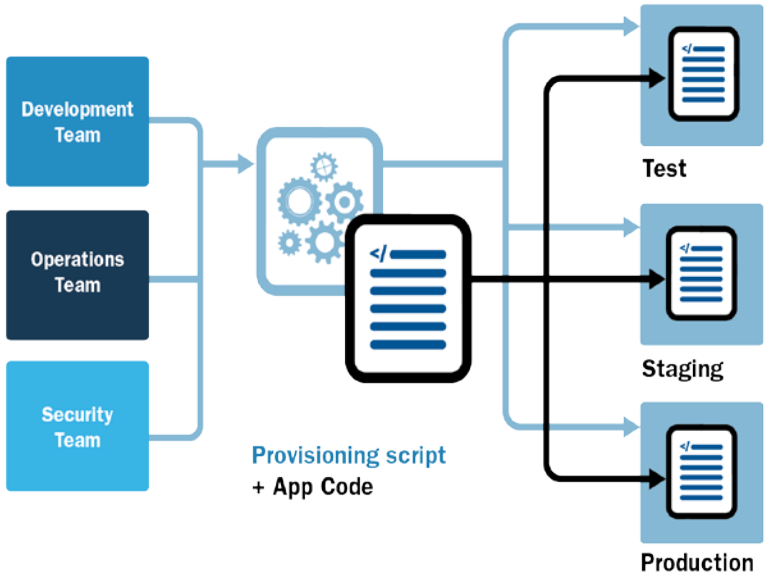
- Constant communication and visible to all
- Ensures that tasks are testable and repeatable
- Frees up human experts to do challenging, creative work
- Allows tasks to be performed with minimal effort or cost
- Creates confidence in task success, after past repetitions
- Faster deployment , frequent quality release

@securelifecycle

DevOps with Agile : *Platform*

Where *people* use *process* to build software

- Automated environment creation and provisioning
- Automated infrastructure testing
- Parity between Development, QA, Staging, and Production environments
- Sharing and versioning of environmental configurations
- Collaborative environment between all stakeholders



@securelifecycle

Achieve Agile Principles with DevOps Techniques

Agile Principle	DevOps techniques
1. Highest priority is satisfy the customer through early and ***continuous delivery*** of software	Continuous Delivery & Deployment
2. Welcome ***changing requirements***, even late in development	Continuous Integration and Continuous Feedback
3. Deliver working software ***frequently***, from a couple of weeks to a couple of months	Continuous Integration Continuous Deployment Continuous Feedback
4. Business people and developers must work together daily throughout the project	Integrated Development Environment

@securelifecycle

Achieve Agile Principles with DevOps Techniques (2/3)

Agile Principle	DevOps techniques
5. Build projects around motivated individuals. ***Provide environment and support*** they need	Infrastructure as Code
6. The most efficient and effective method of conveying information to and within a development team is ***face-to-face conversation***	Communication and collaboration
7. ***Working software*** is the primary measure of progress	Continuous Integration Continuous Deployment
8. Agile processes promote sustainable development. The Sponsors, developers and users should be able to maintain a c***onstant pace indefinitely***	Continues Integration Continuous Delivery Continuous Deployment Continuous Feedback

@securelifecycle

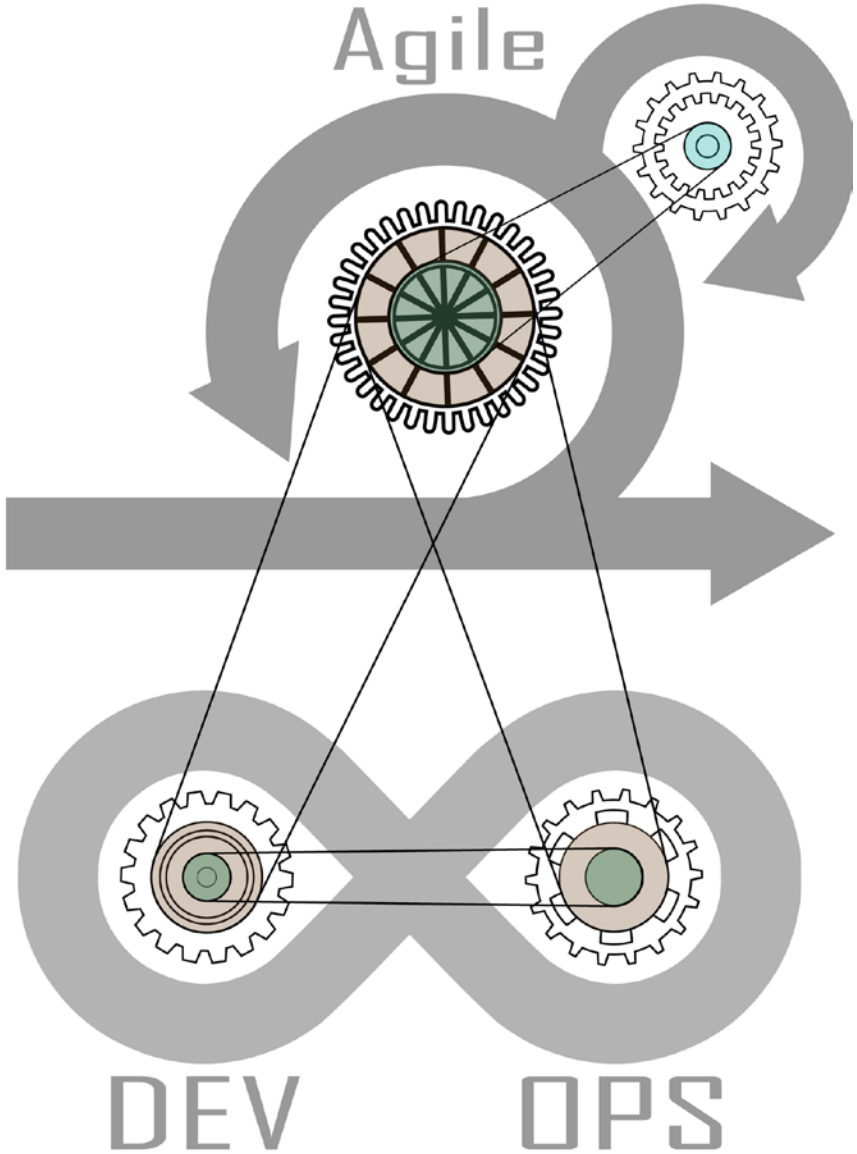
Achieve Agile Principles with DevOps Techniques (3/3)

Agile Principle	DevOps techniques
9. Continuous attention to technical excellence and good design enhances agility	Collaboration and Continuous Integration
10. ***Simplicity—the*** art of maximizing the amount of work not done—is essential	Integrated Automated Development Environment Continuous integration
11. The best architectures, requirements, and designs emerge from ***self-organizing teams***	Communication, collaboration and sharing
12. At regular intervals, the team reflects on how to become more effective, then ***tunes and adjusts*** its behavior accordingly.	Monitoring and measurement

@securelifecycle

Agile Automation with DevOps

Application delivery through collaboration and communication between developers and IT operations with Agile methodologies



@securelifecycle

Topics

- Background
- Agile Manifesto + Methodologies
- DevOps: Reduce time to market
- Barriers on Agile methodologies in large projects
- How DevOps will overcome?
- *With or Without Agile?*
- Conclusion

@securelifecycle

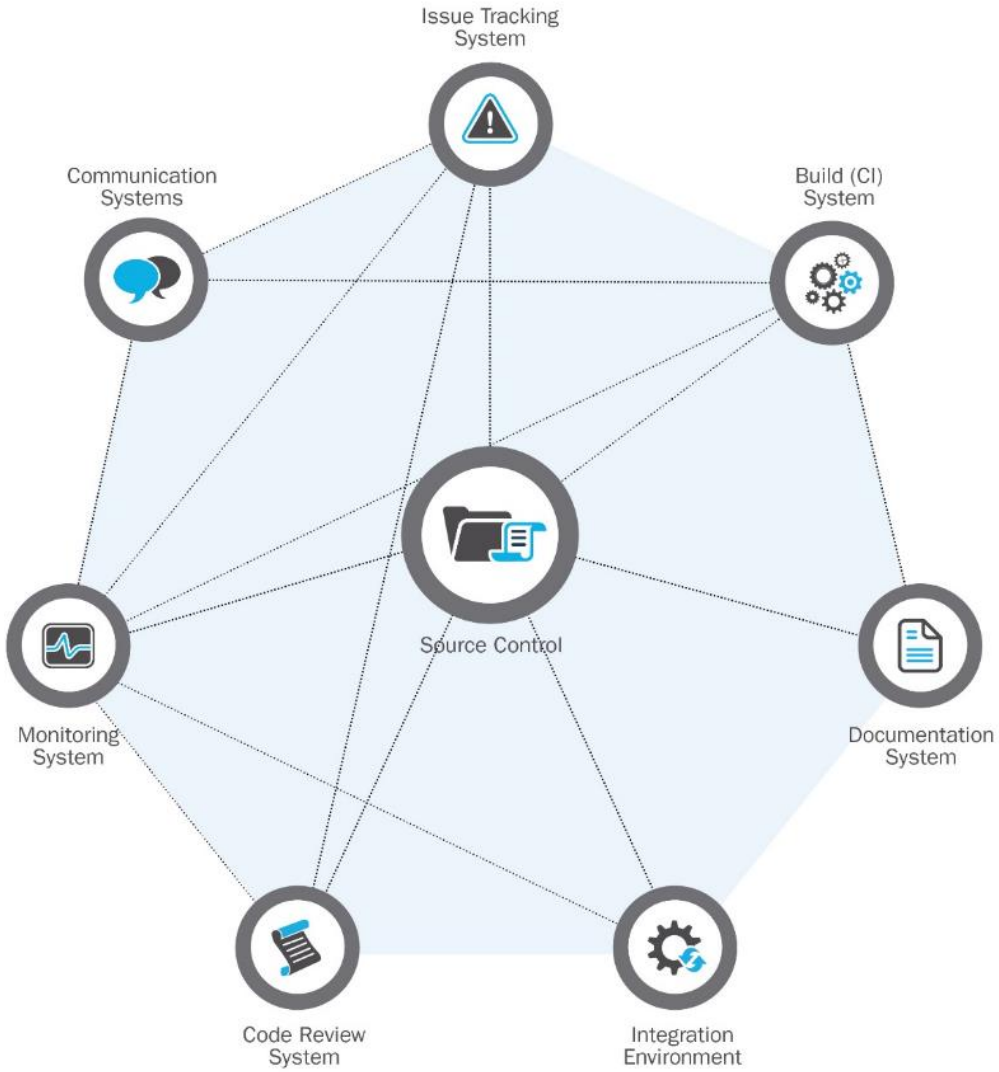
Integrated – Automated Development pipeline

- Modern software development requires integrated deployment pipeline with following characteristics

  - Integrate-ability
  - Interoperability
  - Usability
  - Portability
  - Resilience
  - Security/Permissions
  - Availability (Error handling)
  - Scalability
  - Performance
  - Modifiability
  - Configurability
  - “Automate-ability” (of manual tasks)
  - “Approvability” (allows for manual approval)
  - Measurability?
  - Other?

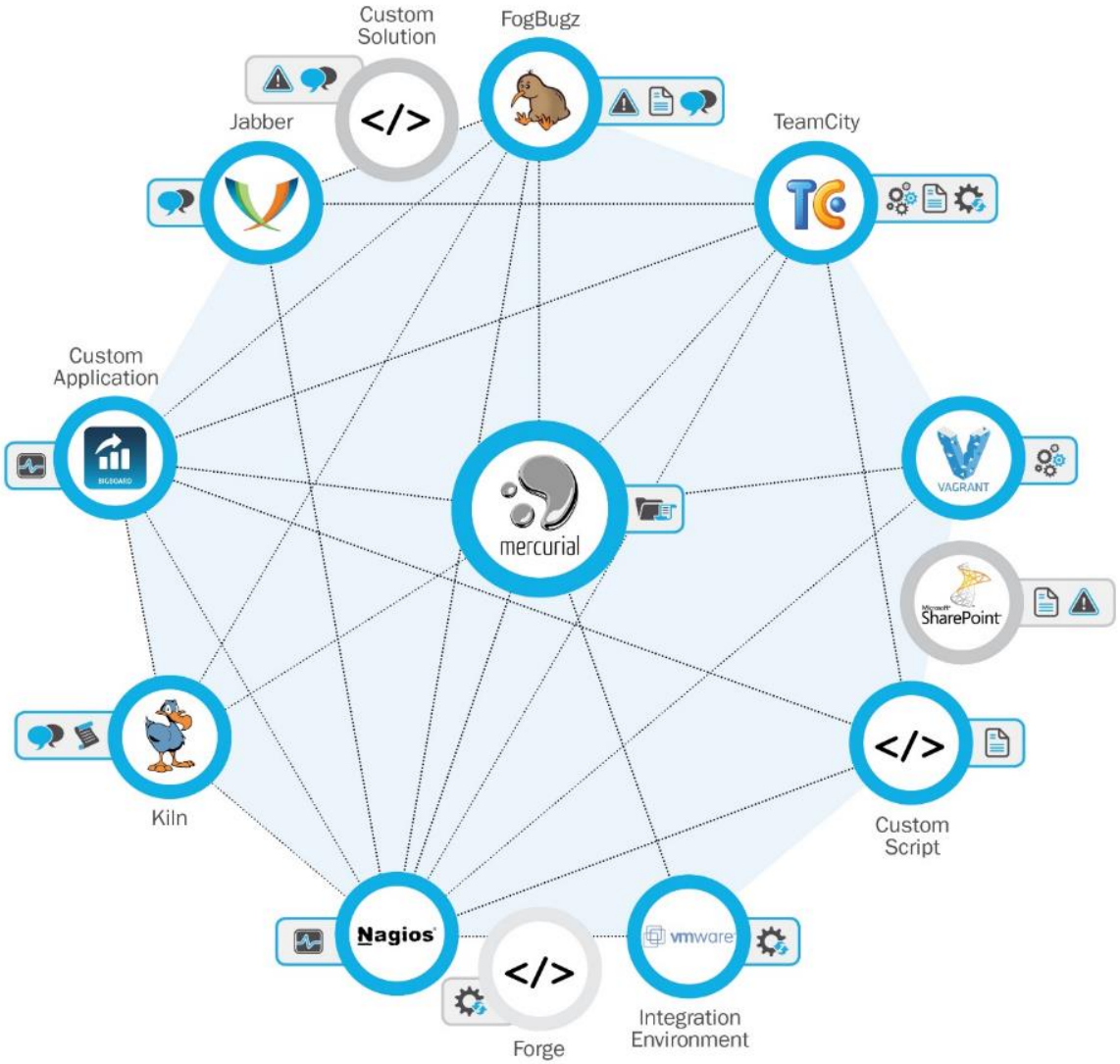
@securelifecycle

Integrated Development Pipeline - General



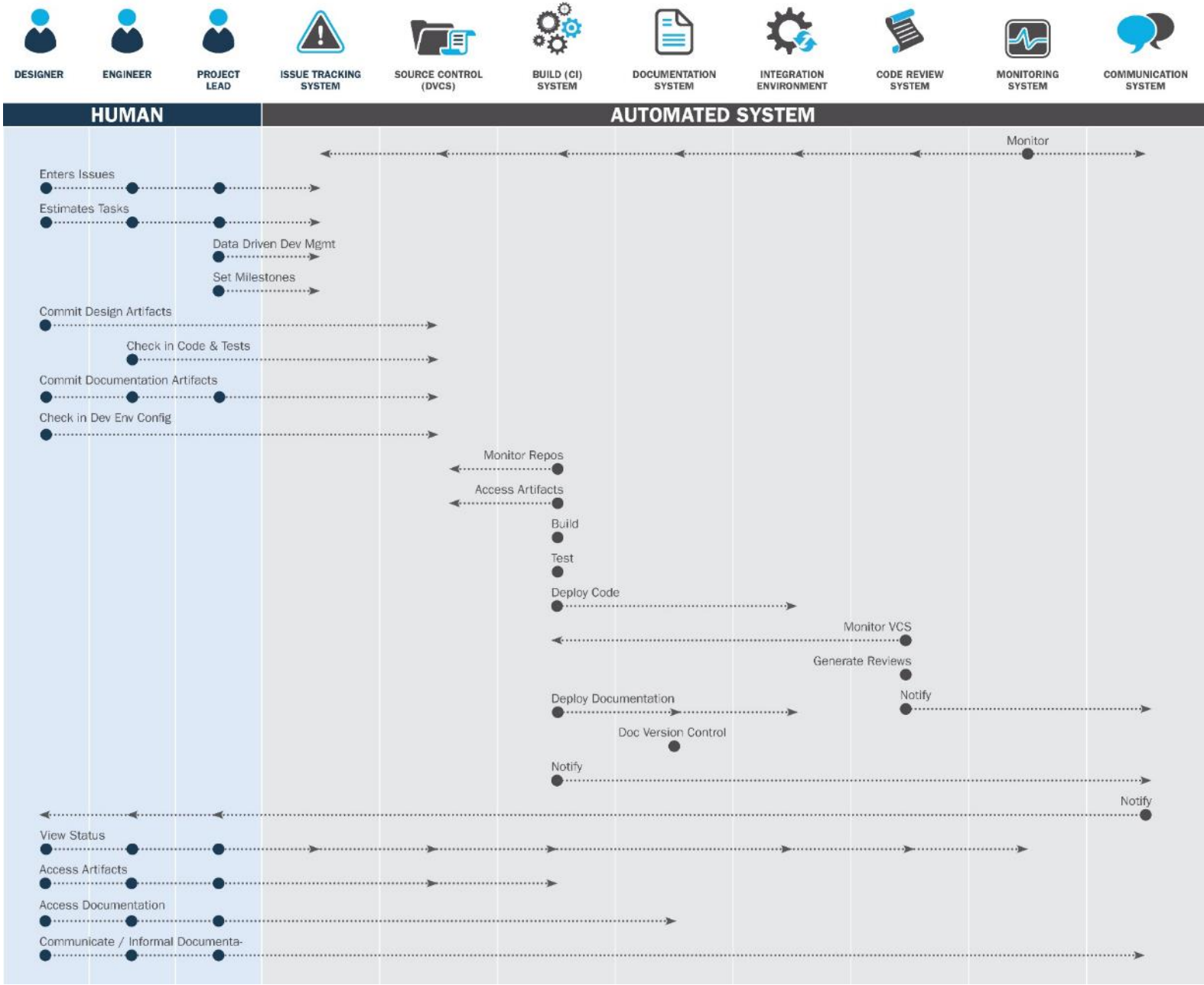
@securelifecycle

Integrated Development Pipeline – With Tooling

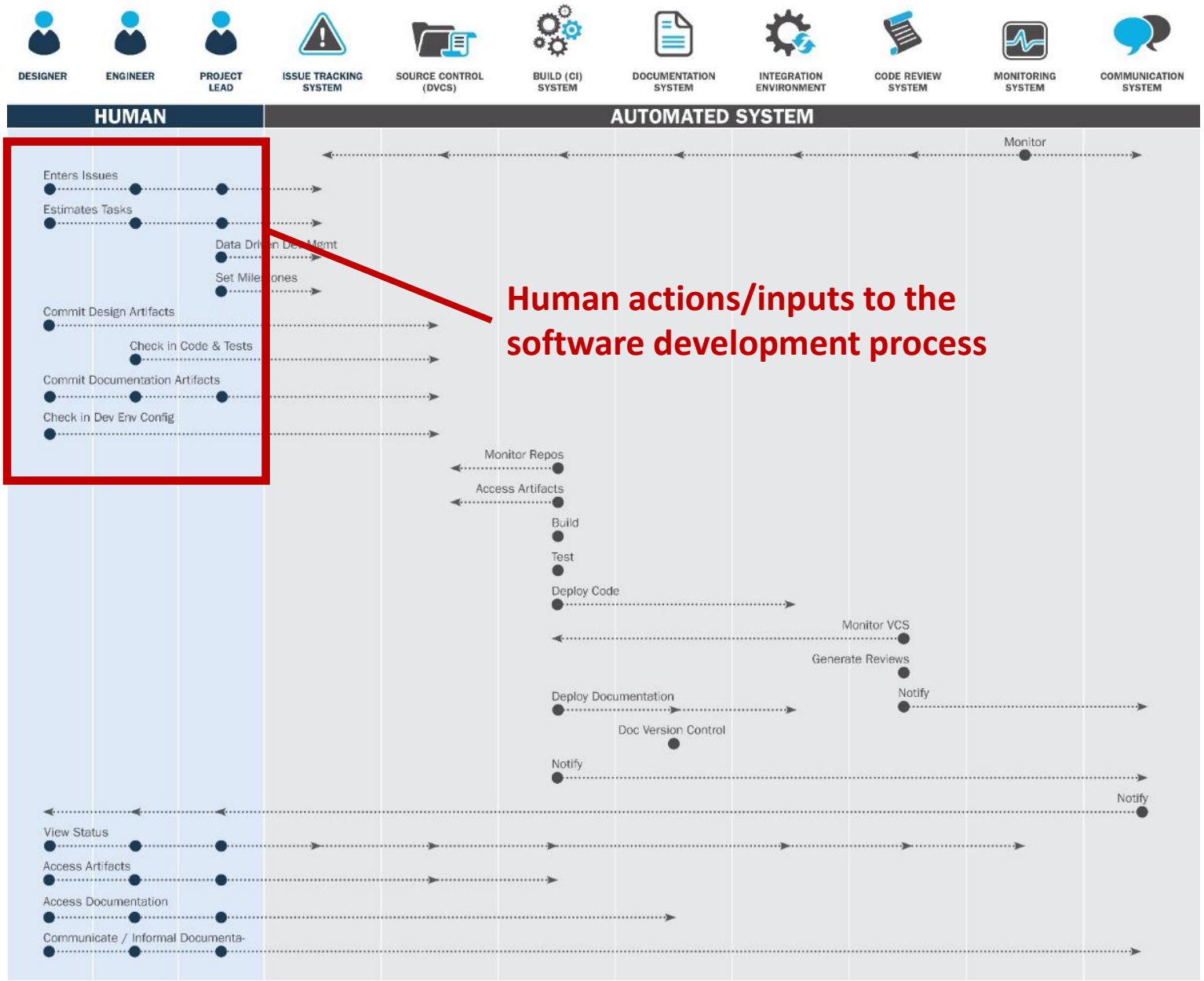


@securelifecycle

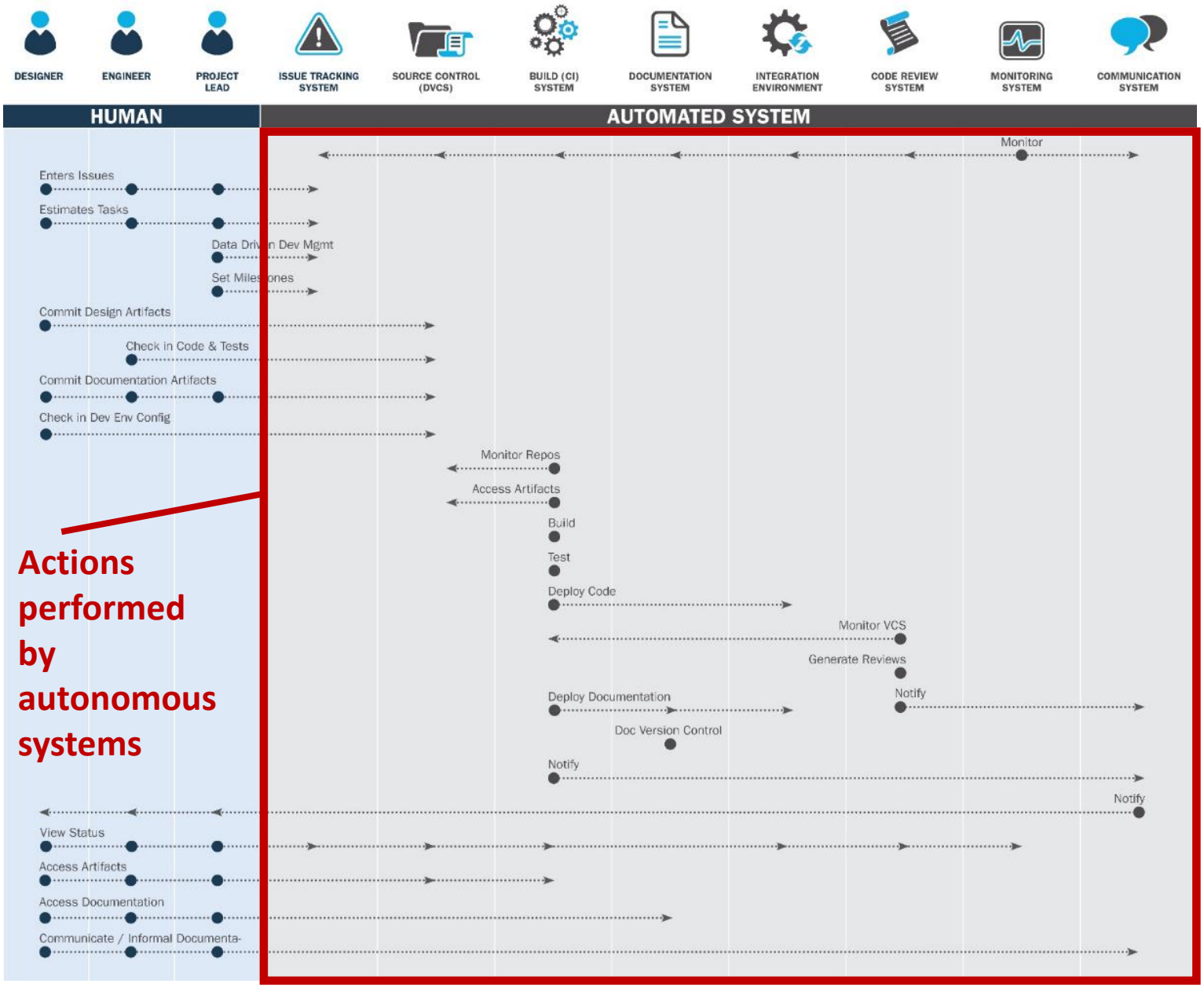
Why Matters?



@securelifecycle



@securelifecycle



@securelifecycle

Application Release Automation (ARA) – Magic Quadrant report

- Future of Modern Development, Automation support on each toolset
- By 2020, 50% of global enterprises will have at least one ARA
- Automation is the feature
- Environment modelling and release coordination as the key requirements
- Support telemetry and apply machine learning to have actionable insights

*Gartner Magic Quadrant for Application Release automation of 2016

@securelifecycle

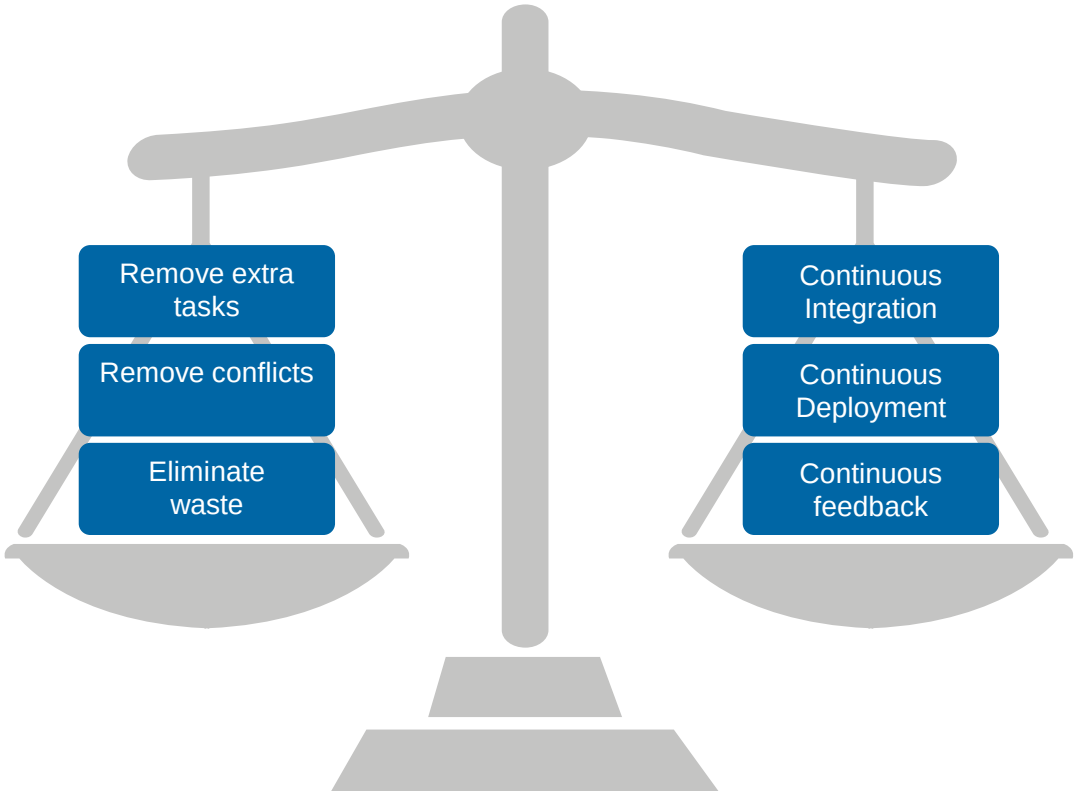
Topics

- Background
- Agile Manifesto+ Scrum
- DevOps: Reduce time to market
- Barriers on Agile methodologies in large projects
- How DevOps will overcome?
- With or Without Agile?
- *Conclusion*

@securelifecycle

Goal: Balance efficiency and effectiveness to deliver the right things right!

With Continuous Everything!



@securelifecycle

Agile ⊂ DevOps

Agile

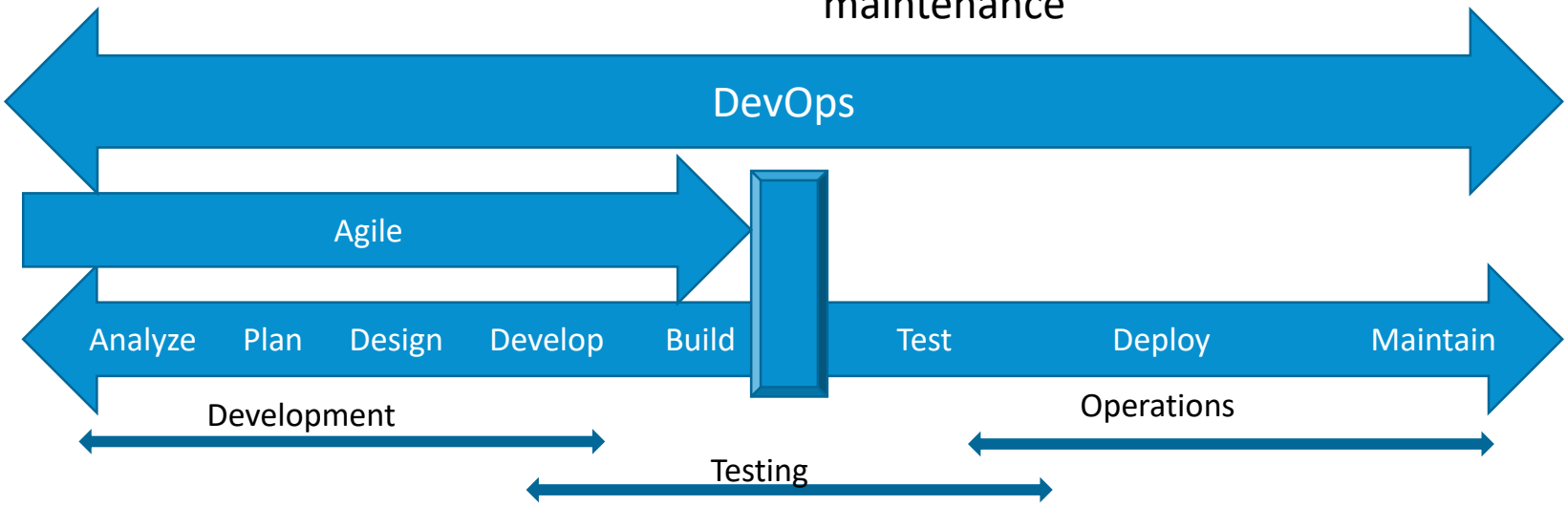
Embrace constant change

Embed Customer in team to internalize expertise on requirements and domain

DevOps

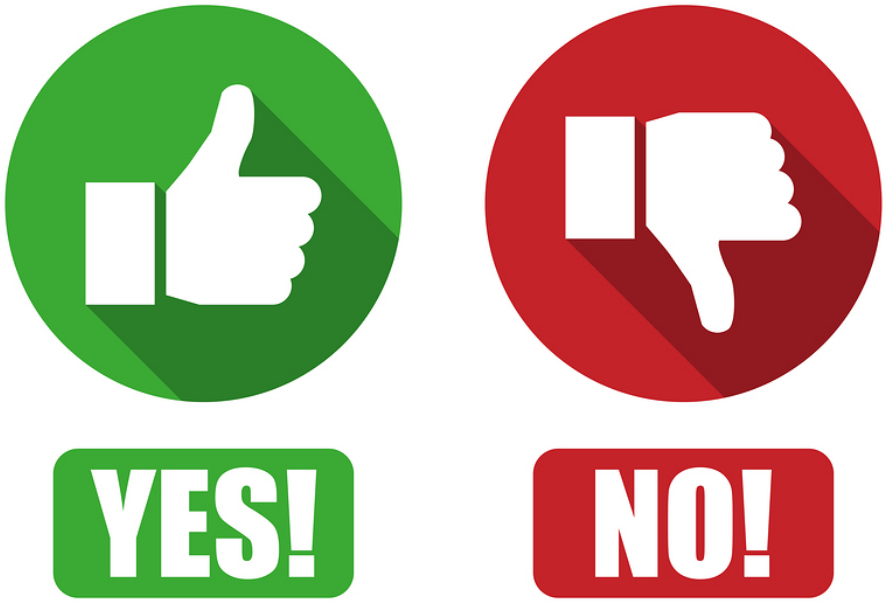
Embrace constant testing, delivery

Embed Operations in team to internalize expertise on deployment and maintenance



@securelifecycle

Can Agile Methodology Survive without DevOps Techniques?



@securelifecycle

SLS team GitHub Projects

- Once Click DevOps deployment
  https://github.com/SLS-ALL/devops-microcosm
- Sample app with DevOps Process
  https://github.com/SLS-ALL/flask_api_sample
  - Tagged checkpoints
    - v0.1.0: base Flask project
    - v0.2.0: Vagrant development configuration
    - v0.3.0: Test environment and Fabric deployment
    - v0.4.0: Upstart services, external configuration files
    - v0.5.0: Production environment
- On YouTube:
  https://www.youtube.com/watch?v=5nQlJ-FWA5A

@securelifecycle

For more information…

SEI DevOps Blog

https://insights.sei.cmu.edu/devops

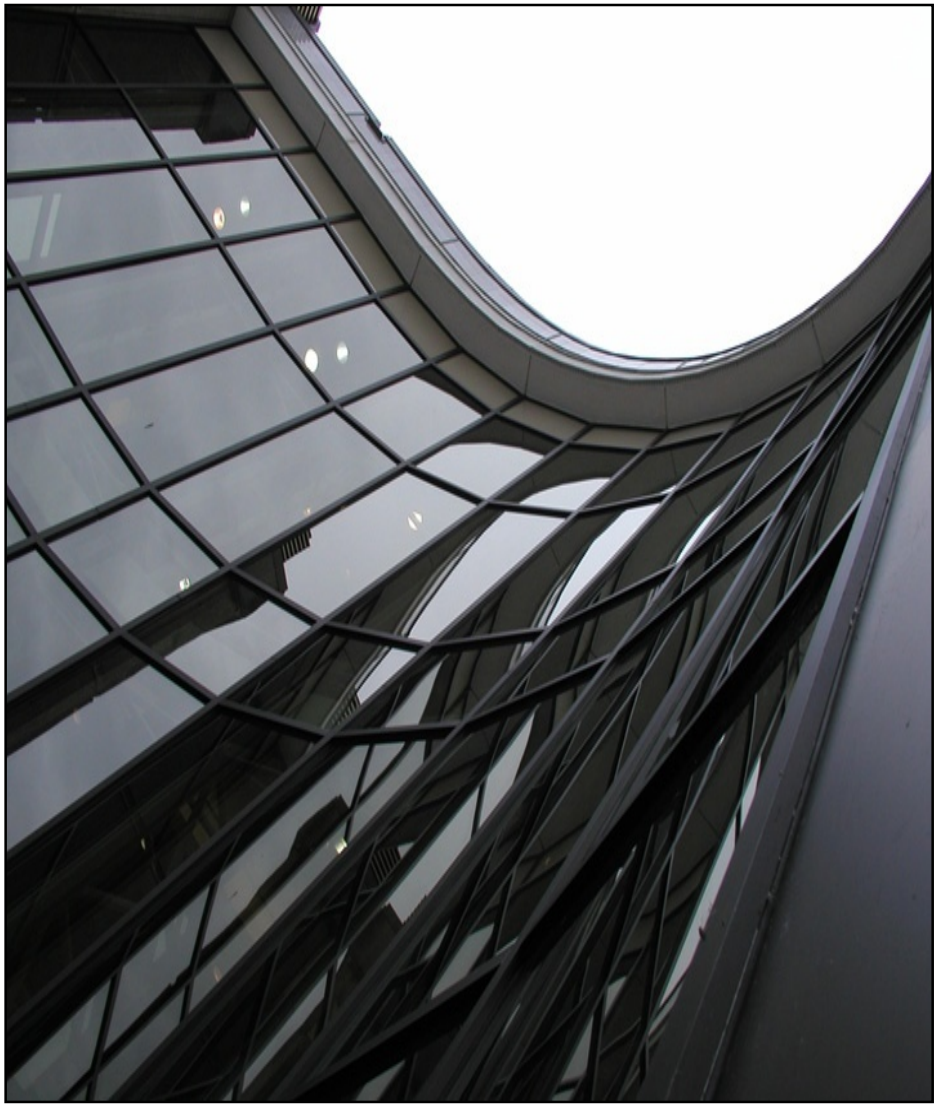
@securelifecycle

Contact Information

Hasan Yasar

Technical Manager,
Secure Lifecycle Solutions
hyasar@sei.cmu.edu

@securelifecycle



Web Resources (CERT/SEI)

http://www.cert.org/

http://www.sei.cmu.edu/