Achieving Digital Transformation(*DX*) with Technical and Cultural Structure

Hasan Yasar

Technical Director, Adjunct Faculty Member
Continuous Deployment of Capability Directorate
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.

DM21-0477

Content Agenda - Expect to discuss :

- **Why?**
  - Source of the problem
- **What?**
  - What is Digital transformation
- **How?**
  - Technical and Cultural Architecture
  - DevOps for System development and deployment
- **Takeaways**

Share my bio

- 25+ years of software development experiences
- Certified Scrum Practitioner
- Certified Ethical Hacker
- Various roles throughout 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
- Faculty Member on delivering DevOps course at CMU, SEI since 2015
- DevOps, DevSecOps community organizer, frequent Speaker
- PC members in various research conferences,
- Editorial board member, IJSS, AJSE
- Member of IEEE & ISO working group on
  - 2675 DevOps standards
  - WG 29 Agile/DevOps

Why Digital Transformation

We live in interesting times

- Physical to Remote
- Everything is connected → Bring complexity + Extensibility
- We all want to transformation! → All values of “US”

The Fabric of Computing is Changing

Networks are becoming software defined

- Generic network nodes can adapt function to usage and demand

Field-Programmable Gate Arrays (FPGAs) are proliferating

- Now even the processor's function is determined by software and malleable after fielding

Emergence of Artificial Intelligence and Machine Learning

- Tasks change from direct programming to data curation and feature discovery

We all really want for our value “US”

- Availability
- Reliability
- Usability
- Observability
- Scalability
- Operability
- Repeatability
- Agility
- ……

All for Free

Without changing anything!

Without change anything!

- **Resistance to change → Transformation is hard**

Without change anything!

- **Resistance to change → Transformation is hard**
  - **We all want more tools…**

Without change anything!

- **Resistance to change → Transformation is hard**
  - **We all want more tools…**
    - **Each tools generates data**

Without change anything!

- **Resistance to change → Transformation is hard**
  - **We all want more tools…**
    - **Each tools generates data**
      - **Disconnected data silos**

Without change anything!

- **Resistance to change → Transformation is hard**
  - **We all want more tools…**
    - **Each tools generates data**
      - **Disconnected data silos**
        - **Forces "US" to be individual teams**

Without change anything!

- **Resistance to change → Transformation is hard**
  - **We all want more tools…**
    - **Each tools generates data**
      - **Disconnected data silos**
        - **Forces "US" to be individual teams**

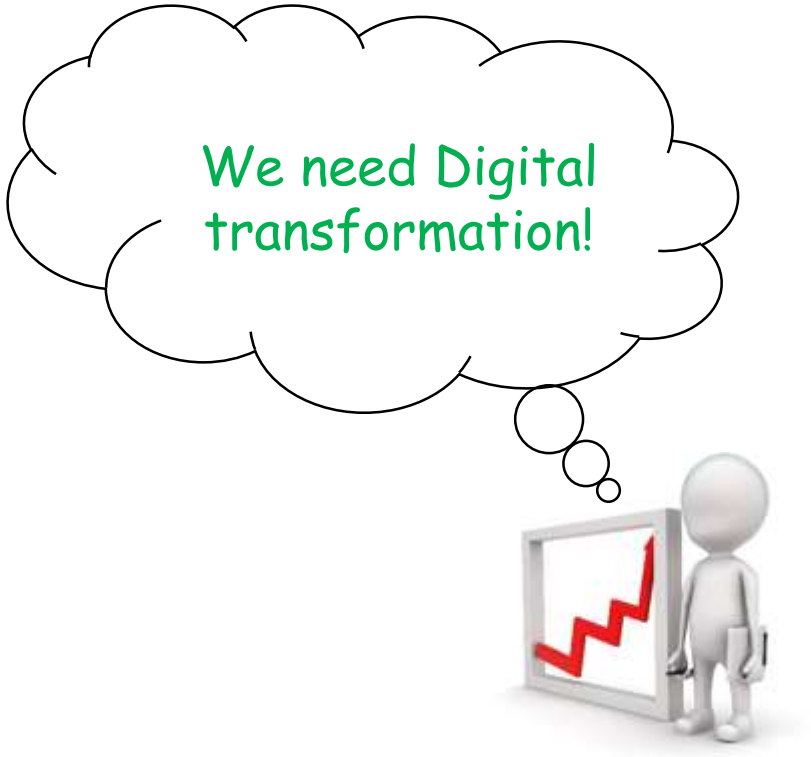
Tool dictates "US" what to do!

Bendor-Samuel, Peter. "Why digital transformations fail: 3 exhausting reasons." Enterpriser's Project, 27 Aug. 2019, https://enterprisersproject.com/article/2019/8/why-digital-transformations-fail-3-reasons.

Main Problems

- Leadership
  - Prevents change
  - Bring bureaucracy
- Product
  - Build things that not accepted by user
  - Not aligned with business strategy
- Development & Architecture
  - Builds wrong things,
  - Delayed on deployments
- Operations
  - Lack of operational excellence (to many incidents and outages)
  - Not able make it informed decision

So what we should do?



What is Digital Transformation(DX)?

“ Digital Transformation” is the adoption of digital technology to transform services or business to meet customer needs”

1. Digitized → Operational Excellence
2. Digital → Rapid Business Innovation

Engaging your customers

Empower your team members

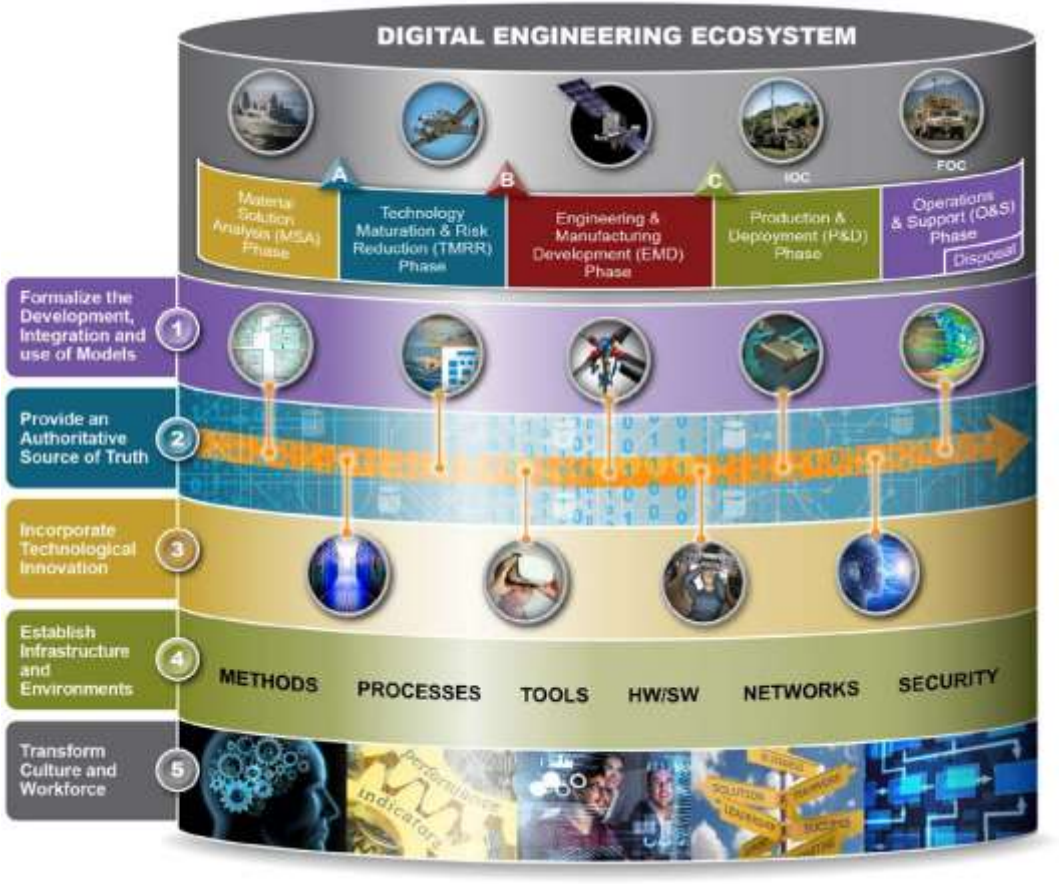
Enable operational excellence

Transform your products

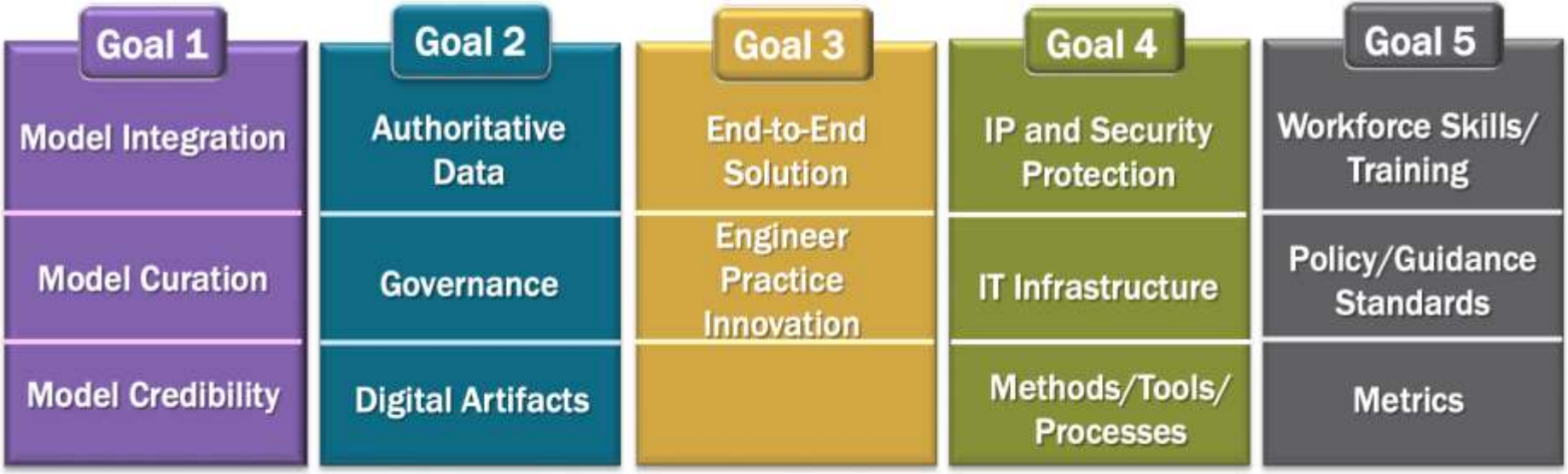
“Digital Engineering”

“digital engineering as an integrated digital approach that uses authoritative sources of system data and models as a continuum across disciplines to support lifecycle activities from concept through disposal”

*DoD Digital Engineering Strategy 2018”



Digital Engineering Strategy Goal

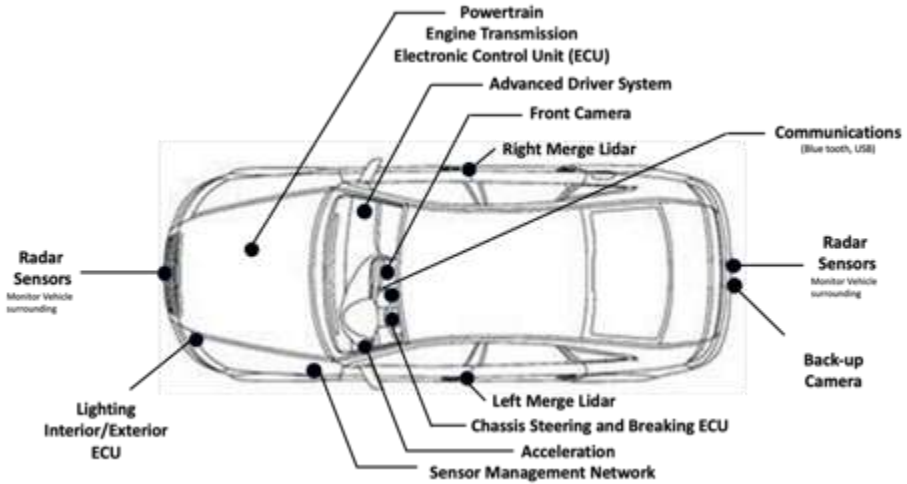


Technical architecture & team organization

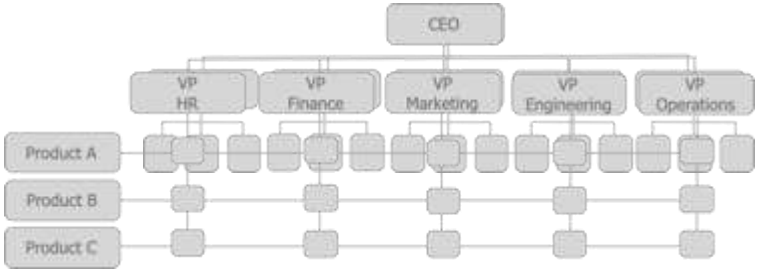
Example: **Autonomous Vehicle**

- Autonomous vehicles have similar complexity and human safety details as many of the products.

System Context

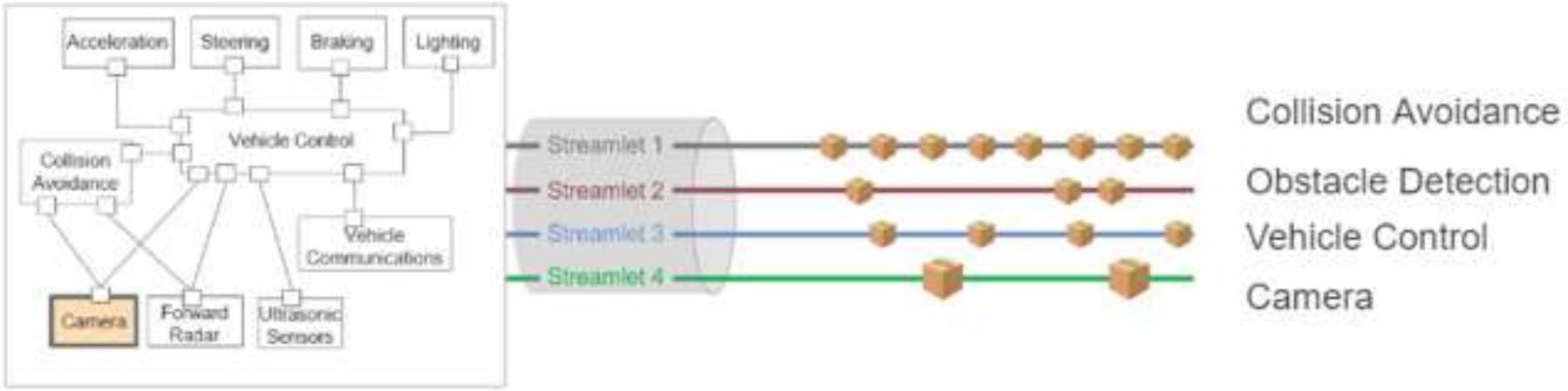


Cultural Context



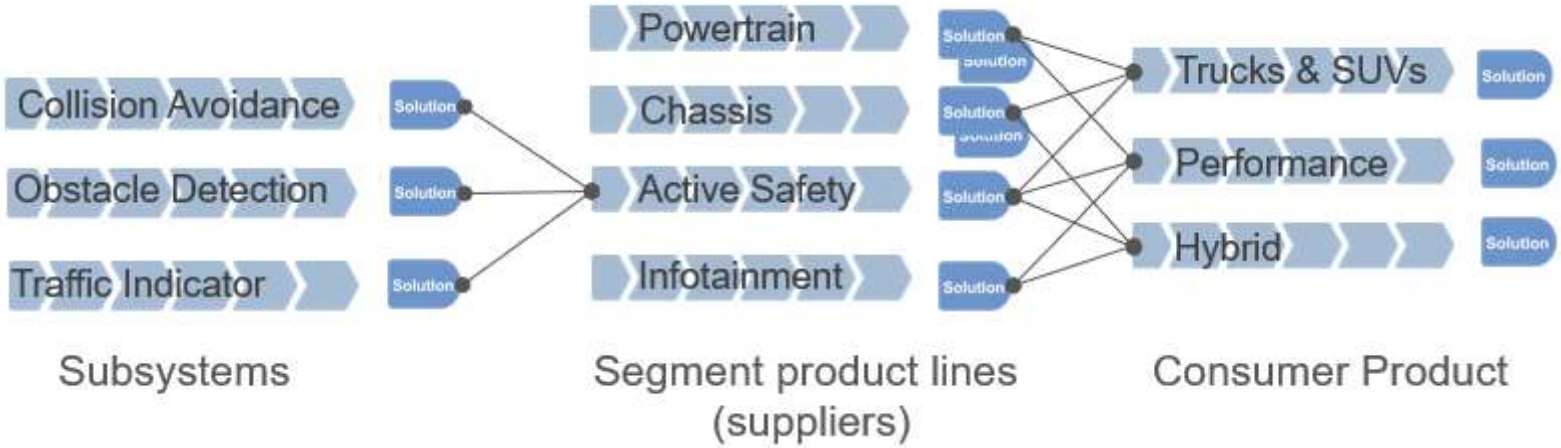
Challenges - Technical Architecture:

- Modularity enables continuous flow in software and hardware.

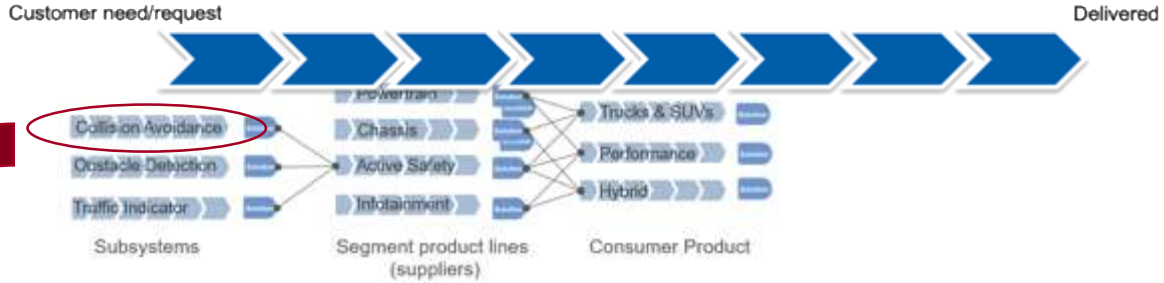


Challenges - Team Organization

- Value streams that produce product/product lines
- Organized around components and functionality

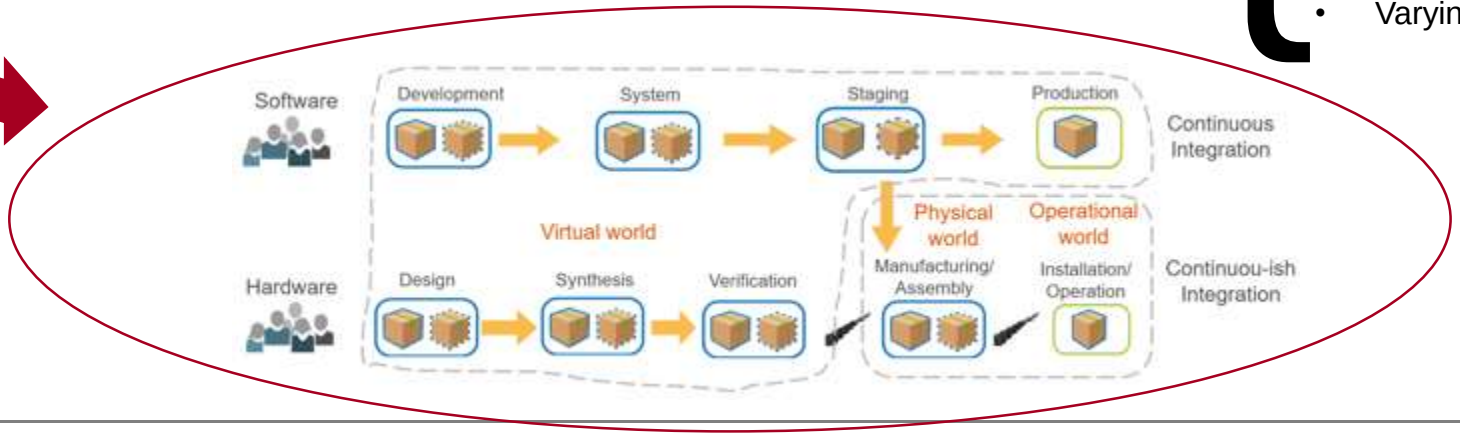


Typical Value Stream Identification (Cultural Context)



Example Autonomous vehicle has several ***"teams"*** to consider

- Organization Structure
- Information Flows
- Heterogenous Subcultures
- Mental Models
- Multiple relationships
- Varying language



Technical Architecture Rolling Roadmap

	Q1	Q2	Q3	Q4
Systems	Validate ”AS IS” state at all system levels against business objectives	Design steel thread to move system to next desired state.	Develop and publish Internal case-studies to share internally (localized outcomes/success)	Provide Real-time business outcome performance visibility
Prototype	Define Prototypes to prove out multiple hypothesis	Leverage Set based design to reduce solution alternatives	Leverage Set based design to reduce solution alternatives	Value Stream based budgeting
Refactor	Use System Telemetry from pipelines to show where improvements are needed	Refactor areas of solution that need to be optimized or simplified.	Teams involved in technical hiring	Align dynamic role-based learning with HR career paths
Innovate	Develop Personas of key customers to understand what future needs they have.	Implement solution day pitches or challenges to continuously evolve.	Leadership team runs as an agile team	Leadership commits to true WIP Limits

Dynamic Learning Feedback Loop

Dynamic Learning Feedback Loop

Dynamic Learning Feedback Loop

Dynamic Learning Feedback Loop

Committed

Forecasted

Intentional Culture Architecture Design

to ease Cognitive Dissonance

Mindset Validation

If the value of change is understood and agreed with, people will be more willing to adjust their mindset and give it a try.

Org Surrounding Support Structure

New behaviors must be enabled by all surrounding structures.

Technical Competency

People must have the technical competency to complete the required action.

Active Role-Modeling

People must see their leaders, peers, and those they respect modeling the change - and the cultural tenets - actively.

Culture Architecture Rolling Roadmap

	Q1	Dynamic Learning Feedback Loop	Q2	Dynamic Learning Feedback Loop	Q3	Dynamic Learning Feedback Loop	Q4	Dynamic Learning Feedback Loop
Mind-sets	DevOps Principles & Mindset overview on “Why” we must change and “What” is it we must do.		Provide relevant external case-studies that drive the point to change		Develop and publish Internal case-studies to share internally (localized outcomes/success)		Provide Real-time business outcome performance visibility	
Structures	Standup Communities of Practice (CoPs) to support		Support permission to fail and innovate with providing awards for failure		Team-based performance awards		Value Stream based budgeting	
Competency	Role based learning– acknowledge the gaps and build learning plans		Brown-bag lunch & learns		Teams involved in technical hiring		Align dynamic role-based learning with HR career paths	
Role-modeling	Leadership commits and uses new language		Leadership participates in DevOps and Lean-Agile leadership book clubs		Leadership team runs as an agile team		Leadership commits to true WIP Limits	

Committed

Forecasted

Important Points to Remember:

- **Breaking down the "wall of confusion" between all stakeholders**
- **Culture of shared responsibility**
- **Collective decision and continuous learning**
- **Iterative and incremental approach**
- **Value based technical architecture that aligns with right culture**

DevOps

What is DevOps?

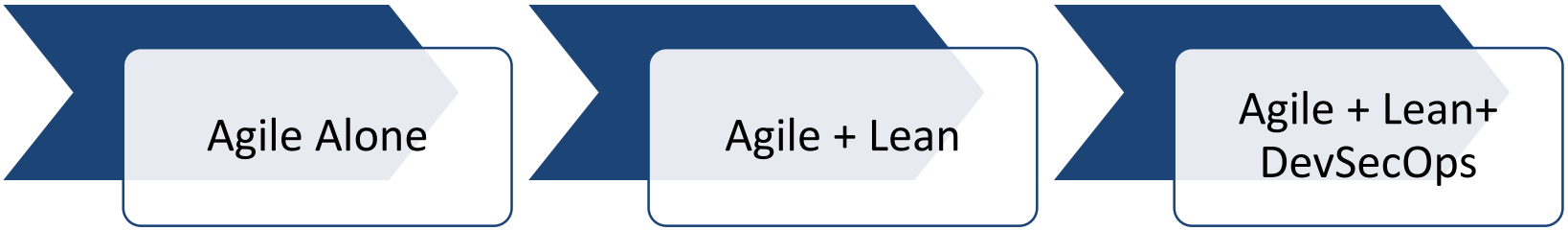
DevOps is a set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing, continuously improving, and operating software and systems products and services [1]

What isn't *DevOps*?

Tools, Tools, Tools

[1] IEEE P2675 DevOps Standard for Building Reliable and Secure Systems Including Application Build, Package and Deployment

Why Agile *AND* Lean *AND* DevSecOps?



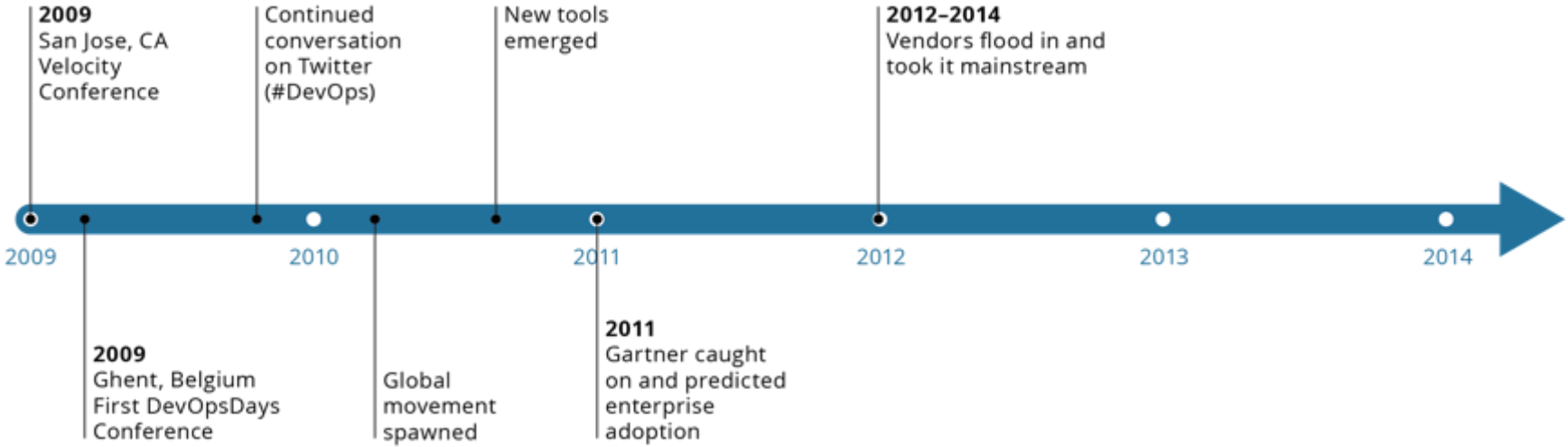
- Tends to focus just on small software teams
  - DOD context typically bigger, more complex
- Tends to assume that direct delivery to customers is feasible

- Adds typical hardware, system, and business/ management teams
- Adds principles and practices that reflect the larger complex system context
- Adds consideration of stakeholders like DT/OT (dev test and opn'l test) and certification

- Adds fast feedback technology infrastructure for continuous architecture, continuous integration, continuous deployment
- Particularly adds to Agile teams' efficiency and effectiveness in execution

DevSecOps is Newer

- The birth of DevSecOps was Patrick Debois's desire for "Agile Infrastructure."
- DevSecOps started as a grassroots movement—of practitioners, by practitioners.
- It caught on, went viral, not because of hype, but because of real results.
- —it's decentralized and open to all.



DevOps Is an Extension of Agile Thinking

Agile

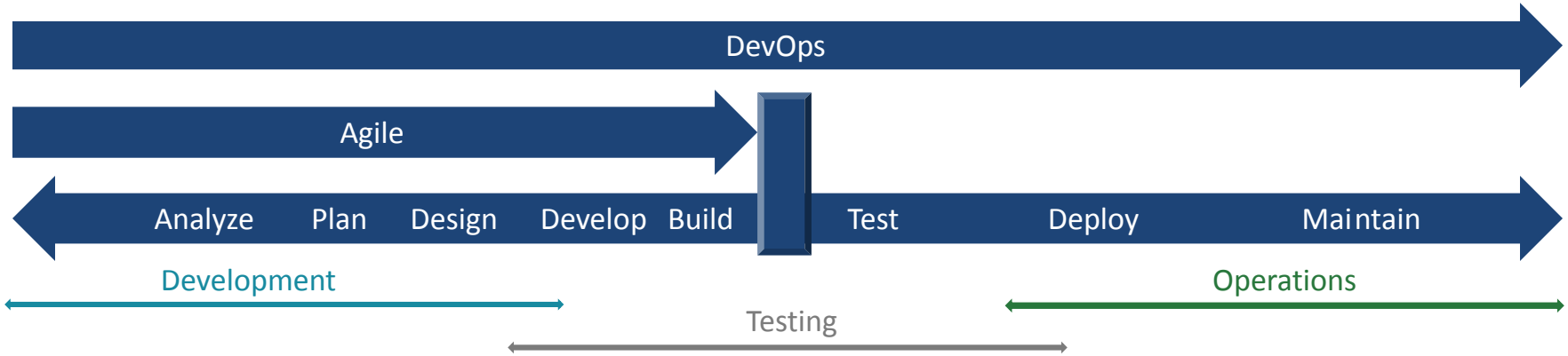
Embrace constant change

Embed customer in team to internalize expertise on requirements and domain

DevOps

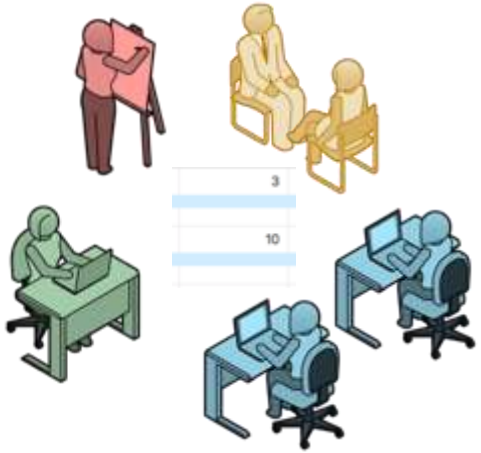
Embrace constant testing, delivery

Embed operations in team to internalize expertise on deployment and maintenance



DevOps is an Extension of Agile Thinking

Agile Team Concept



Oct 1	2	3	4	5
Sprint 2				
8	9	10	11	12
2 more...				• Demo Day

- Cross-functional team agrees to what can be accomplished in a sprint
- Teams are incentivized to help each other
- The team provides a demo to the customer at the end of the sprint
- Customer and leadership can correct course during next sprint

Silos Reinforce the Waterfall Hierarchy

In the waterfall hierarchy, a team finishes a task and tosses it over to the next team. Silos ensure there is no cooperation between teams.

Analysts

Requirements

System Engineers

Design

Developers

Implementation

Testers

Verification

IT Operations

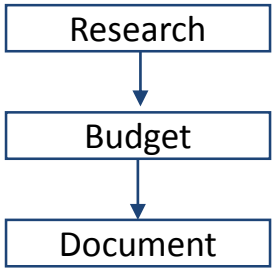
Maintenance

Waterfall structures make it costly and difficult to adapt to change.

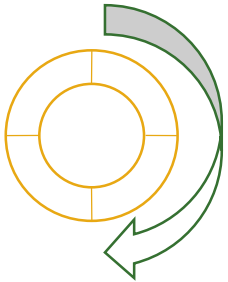
While many groups attempt Agile, they often end up using Water, Scrum, Fall tactics instead.

Water - Scrum - Fall

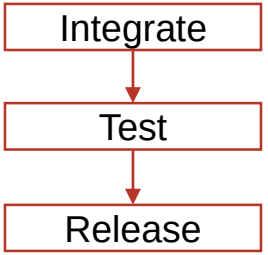
Business



Development

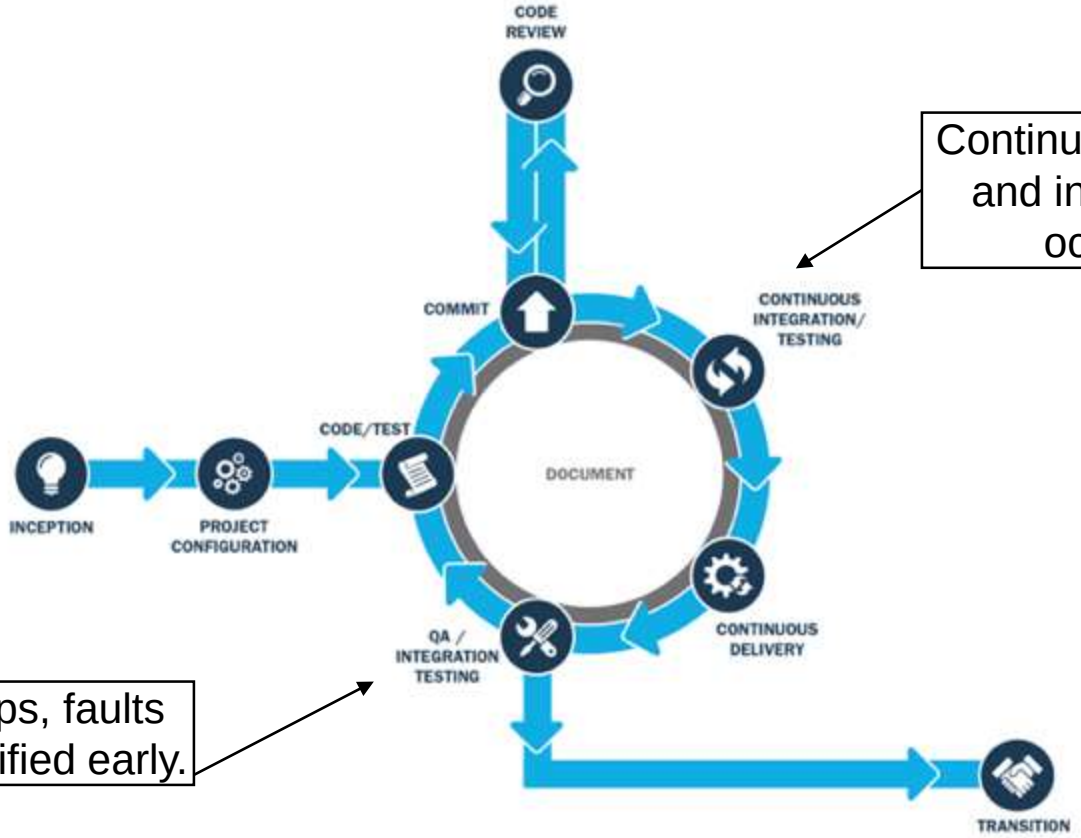


QA Operations



Jez Humble, “Lean Configuration Management,” ChefConf 2015, https://youtu.be/L1w2_AY82WY
Dave West, “Water-Scrum-Fall Is the Reality of Agile,” 2011, http://sdtimes.com/analyst-watch-water-scrum-fall-is-the-reality-of-agile/

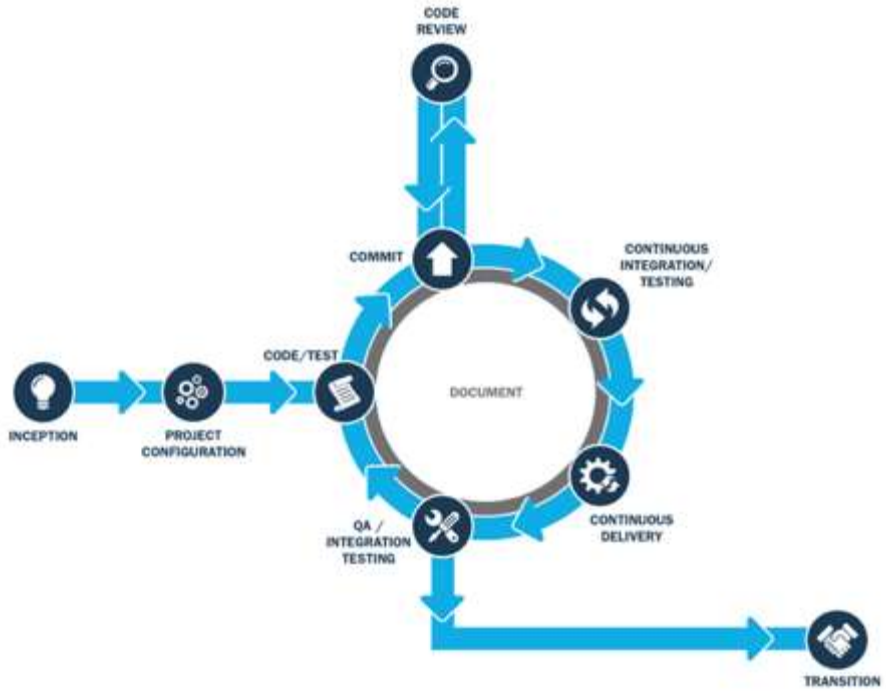
DevOps Team



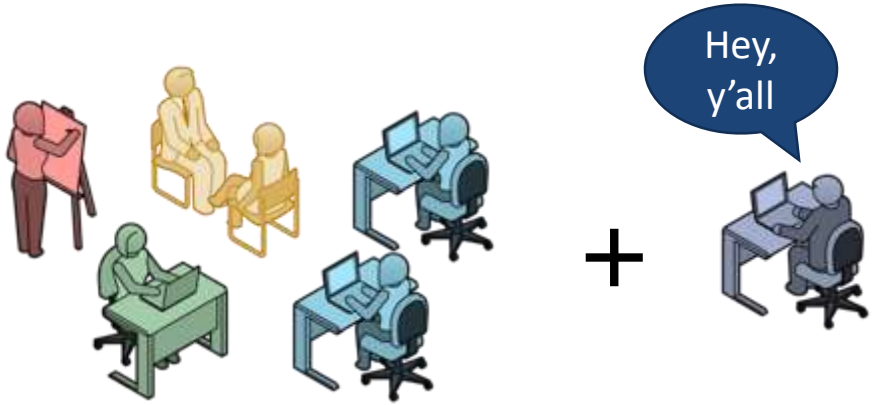
Continuous testing and integration occurs.

With DevOps, faults can be identified early.

DevOps Team



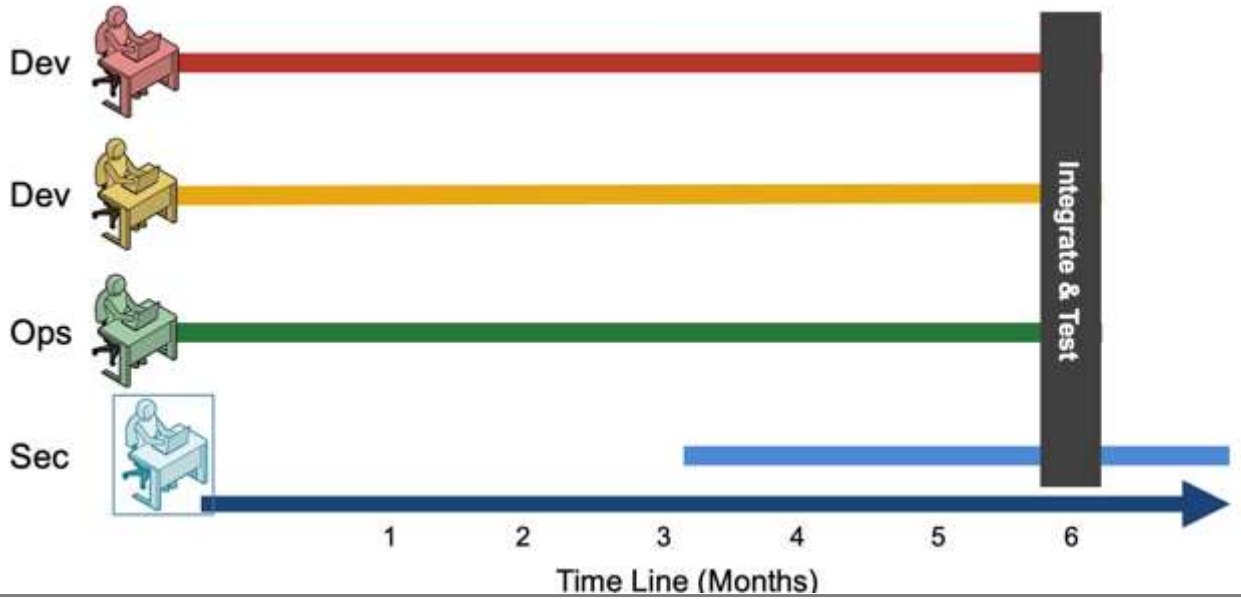
Cross-Functional Dev Team, Including IT Operations



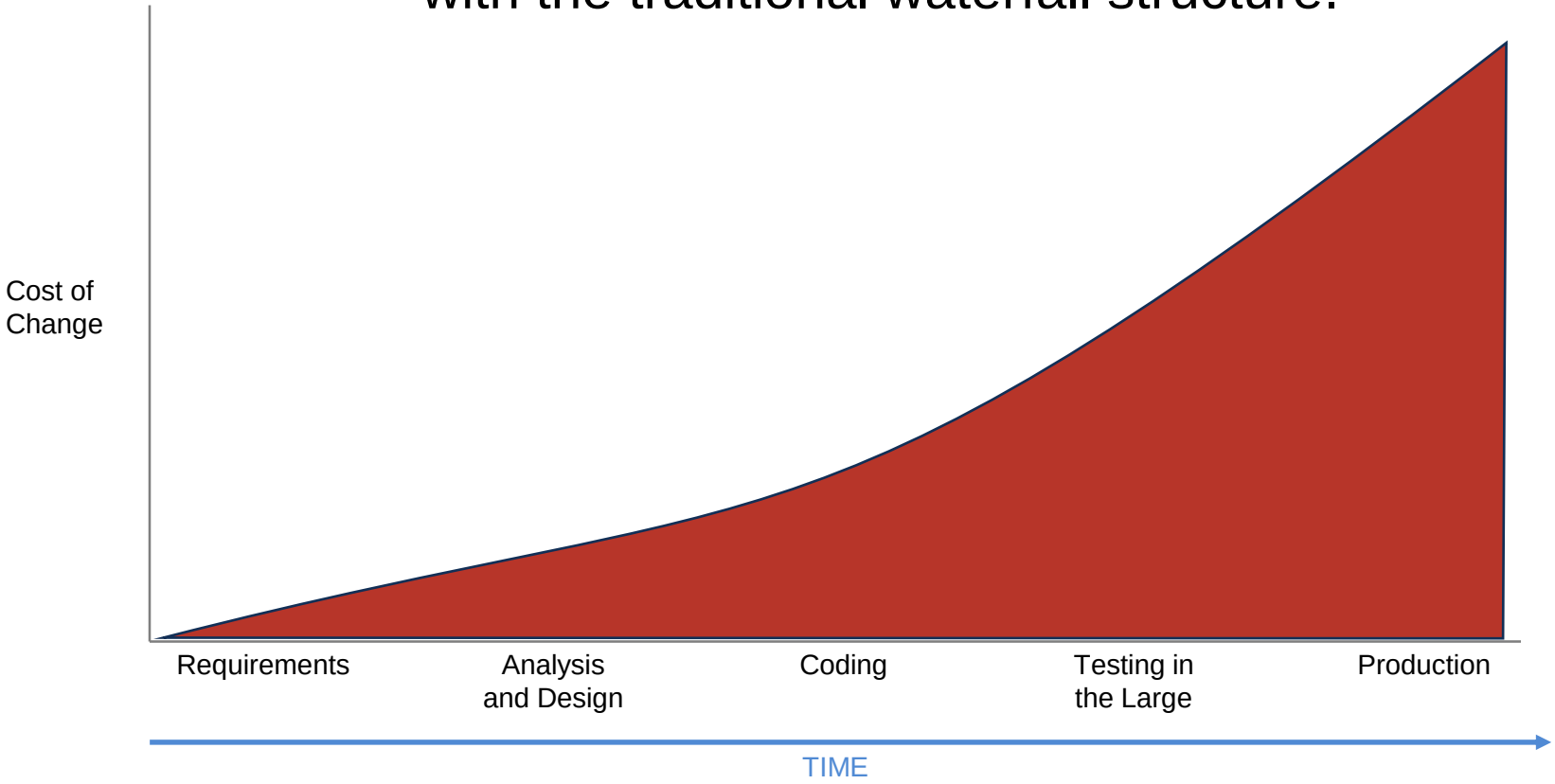
- IT Ops included early in development – deploy to ops-like environment EARLY
- Automation enables fast testing and deployment

Why This Matters: Waterfall Timeline Complications

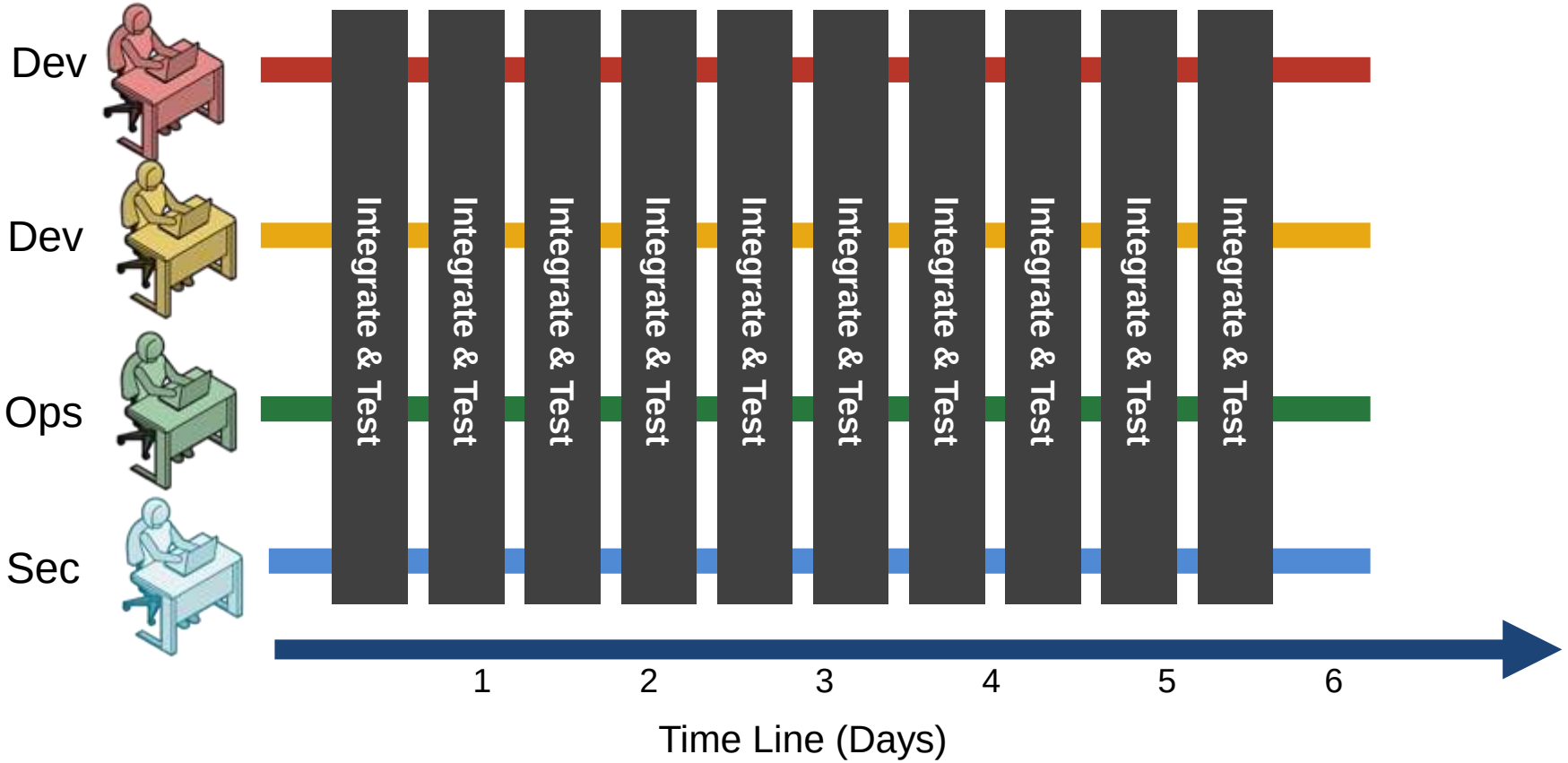
In a waterfall scenario, integration and testing only occurs at the end of the timeline after months of Development / Operations / Security work.



The cost of change increases **exponentially** over time with the traditional waterfall structure.

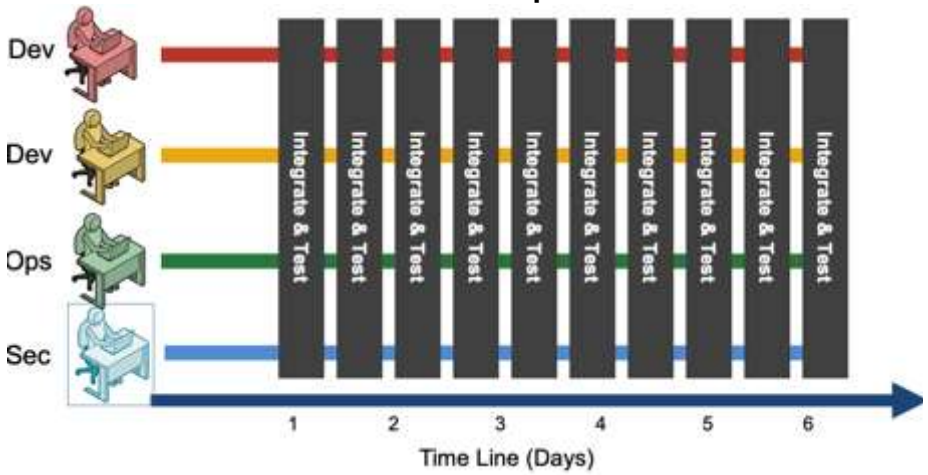


Why This Matters: DevOps Timeline

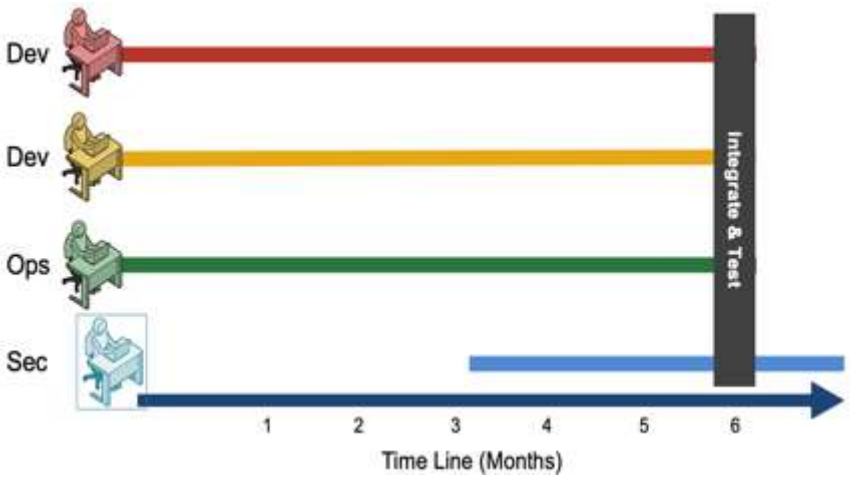


Why This Matters:

DevOps



Waterfall Hierarchy



Switching to adopt DevOps practices can reduce the current required time to integration and testing from months to days.

DevOps ROI

Yearly Returns Possible from Cost of Unnecessary Rework Avoided

Technical Staff Size	x	Average Salary	x	Benefits Multiplier	x	Percentage of Time Spent on Unnecessary Rework	=	Cost of Unnecessary Rework Avoided per Year

	High IT Performer	Medium IT Performer	Low IT Performer
LARGE ORGANIZATION that relies on in-house software (8,500 technical staff)	8,500 staff x $105,000 salary x 1.5 benefits x 1% rework = **$13.4M**	8,500 staff x $105,000 salary x 1.5 benefits x 12% rework = **$160.7M**	8,500 staff x $105,000 salary x 1.5 benefits x 7% rework = **$93.7M**
MEDIUM TO LARGE TECHNICAL ORGANIZATION (2,000 technical staff)	2,000 staff x $105,000 salary x 1.5 benefits x 1% rework = **$3.2M**	2,000 staff x $105,000 salary x 1.5 benefits x 12% rework = **$37.8M**	2,000 staff x $105,000 salary x 1.5 benefits x 7% rework = **$22.1M**
SMALL TO MEDIUM BUSINESSES AND NON-TECHNICAL ENTERPRISES (250 technical staff)	250 staff x $105,000 salary x 1.5 benefits x 1% rework = **$393.8K**	250 staff x $105,000 salary x 1.5 benefits x 12% rework = **$4.7M**	250 staff x $105,000 salary x 1.5 benefits x 7% rework = **$2.8M**

*DORA, DevOps ROI

DevOps Has Four Fundamental Principles

1. **Collaboration:** creating ‘cross-functional’ teams

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

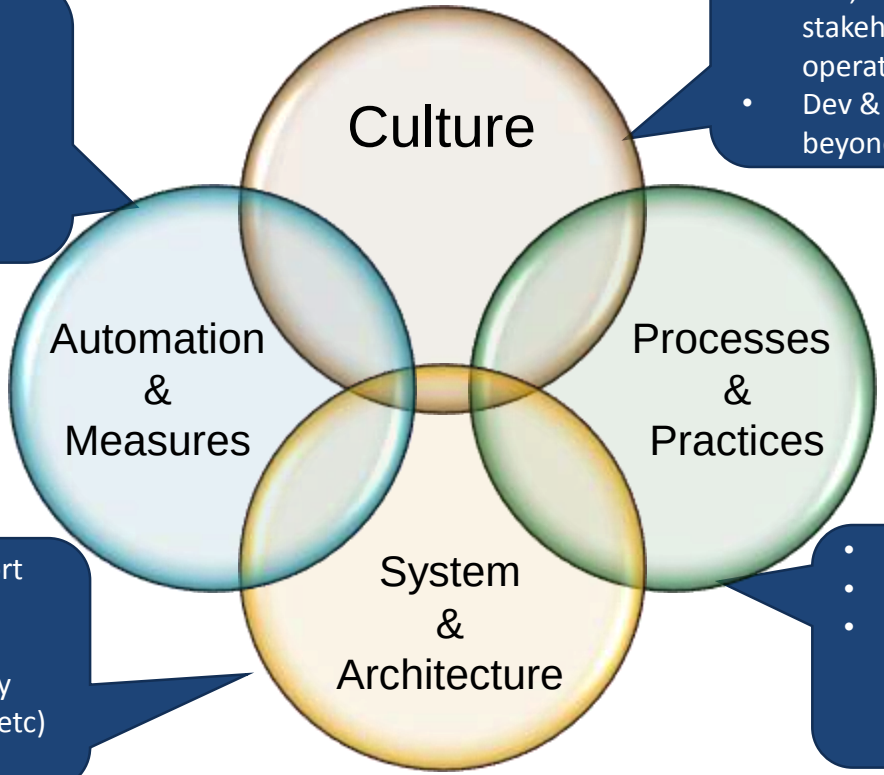
3. **Automation:** deployment, testing, provisioning, any manual or human-error-prone process

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

Might Seem Simple, but not EASY!

- What Some People Think Boundaries of DevSecOps is!
- Automate repetitive, error-prone tasks
- Static & Dynamic Systems Analysis
- Performance dashboards

- All roles collaborate
- Dev, Ops, Sustainment have stakeholders that understand operational drivers
- Dev & Ops support products beyond delivery



- System architected to support integration and automation goals
- Represents important quality attributes (scalable, secure, etc)

- Value stream understanding
- Whole pipeline accounted for
- Continuous integration, automated test, virtualization, self-serve, scripting, automated deployment…

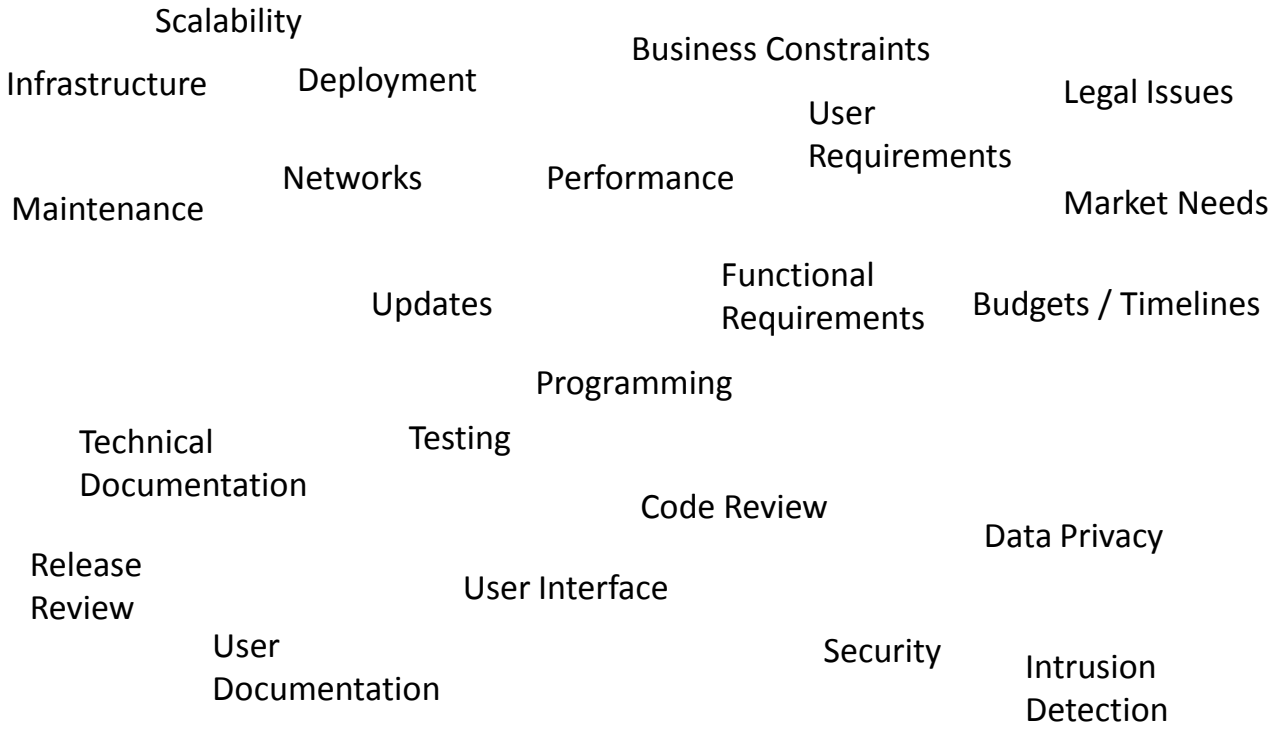
Agile practices+ *Architecture* + *Process*+ *Culture* = **Requirements** for an *Automated* DevSecOps Pipeline

The DevSecOps technology stack is *NOT* where we start

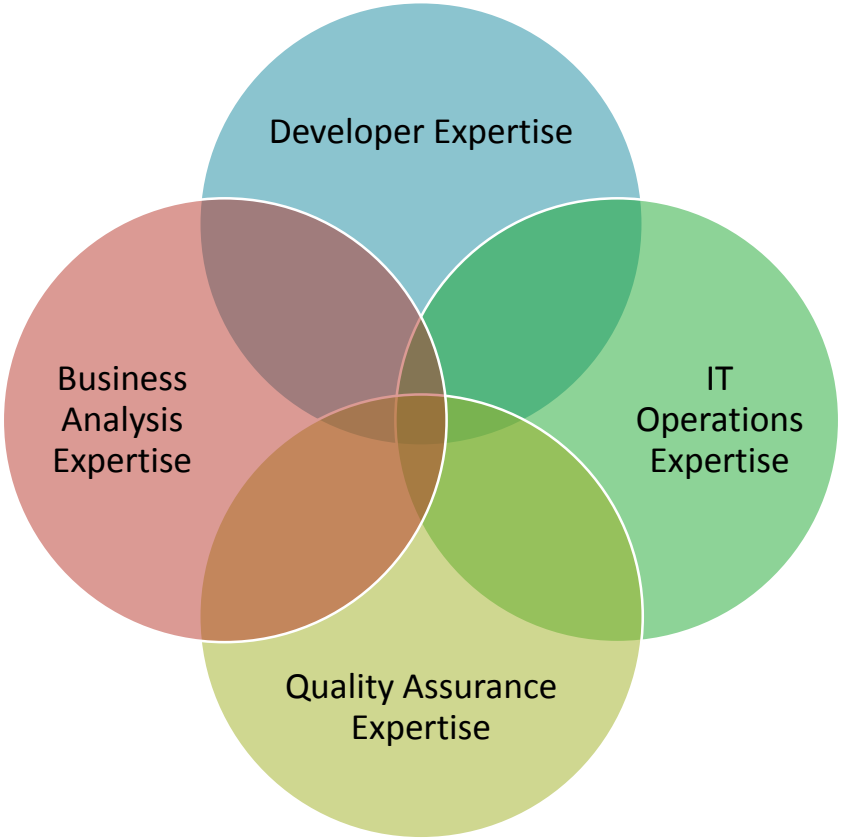
- Automating processes that don't add to or lead to value is more waste

DevOps Foundations:
Communication and Collaboration for sharing

Software Projects require a complex mix of Teams

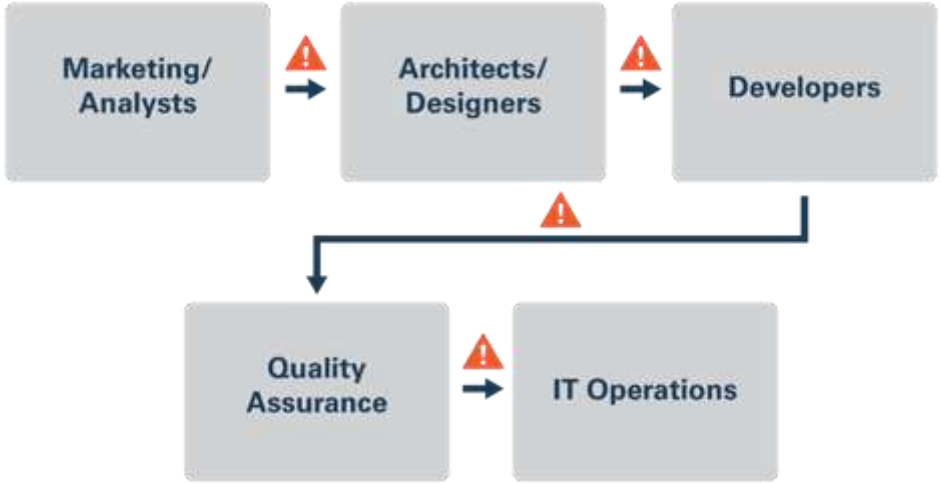


DevOps requires the Collaboration of all of teams



Without the collaboration of these components, issues are only caught at the end of the process.

Every Transition of the System is a Risk



DevOps promotes Collaboration and Continuous Feedback

Heavy collaboration between Dev and Ops on:

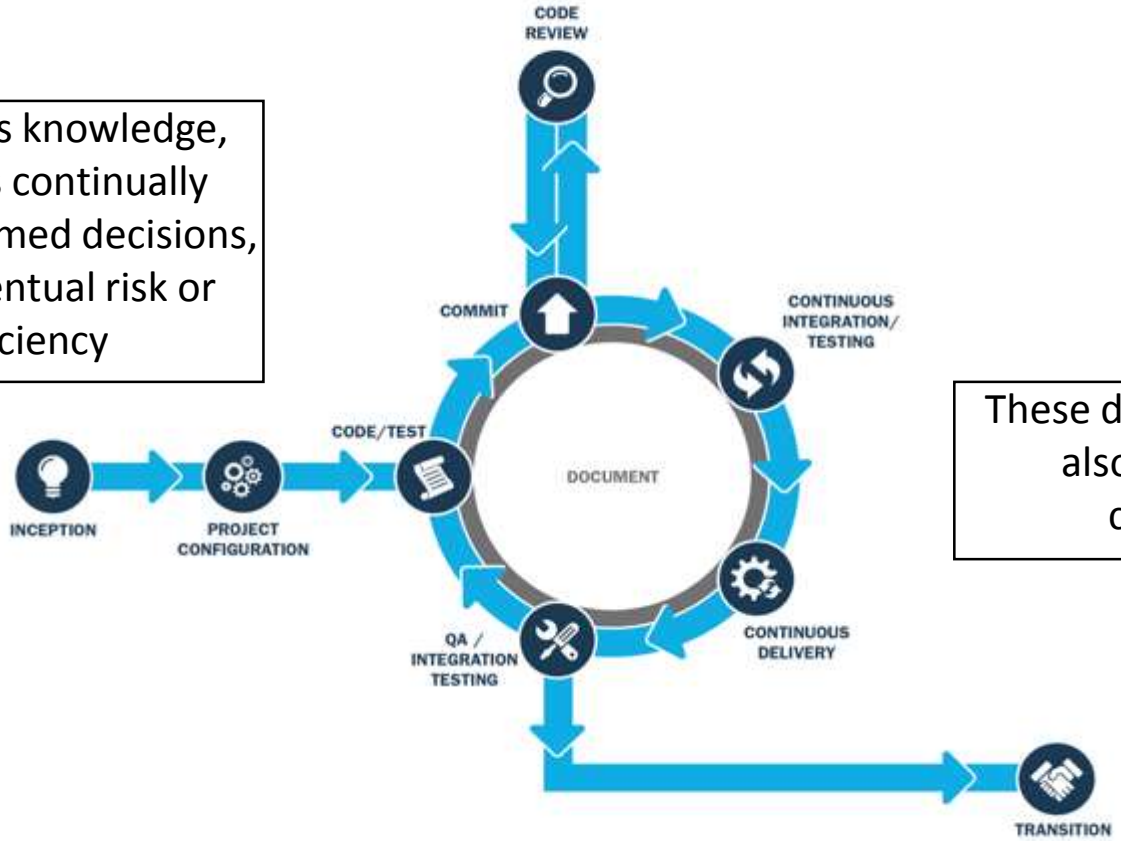
- Design / Architecture decisions
- Environment / Network configuration
- Deployment planning
- Code Review

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

A Software Development Life Cycle (SDLC) is full of decision points

Without Ops knowledge, developers continually make uninformed decisions, causing eventual risk or inefficiency



These decision points are also heavily time consuming.

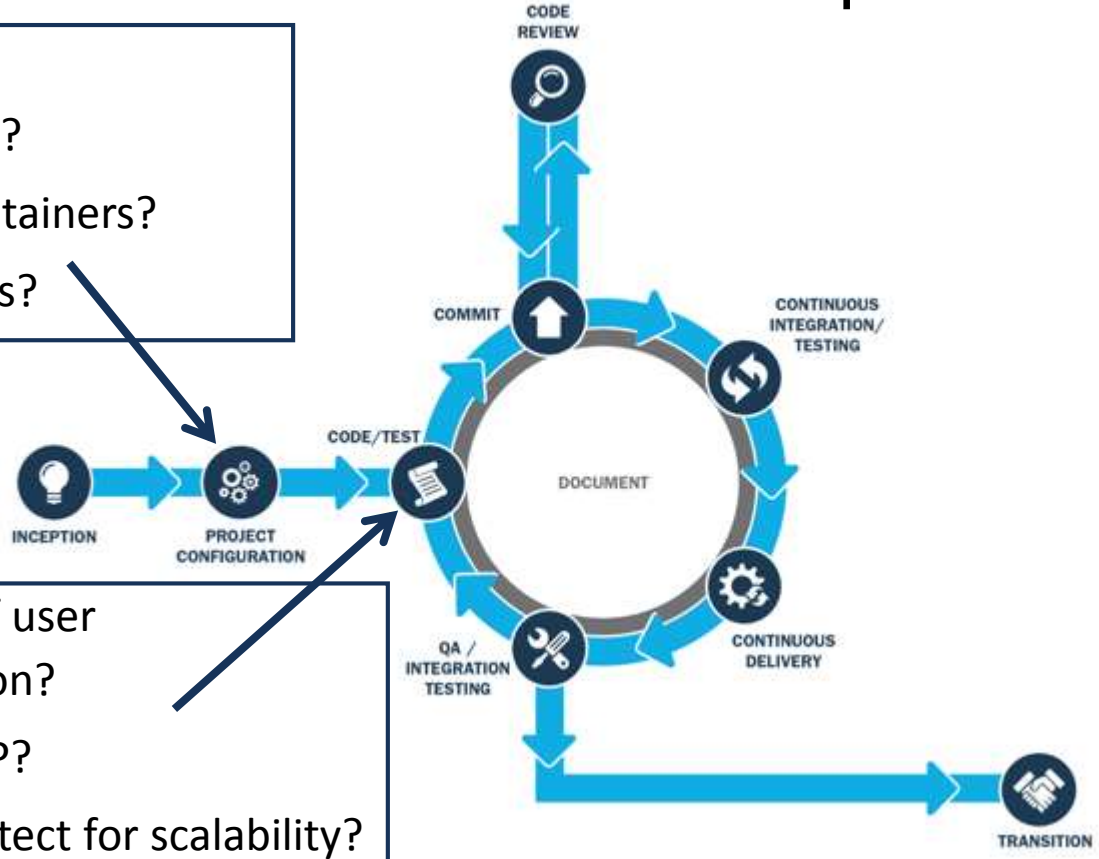
SDLC Decisions Have Architectural Implications

What OS?

What ports?

VMs or containers?

Entry points?



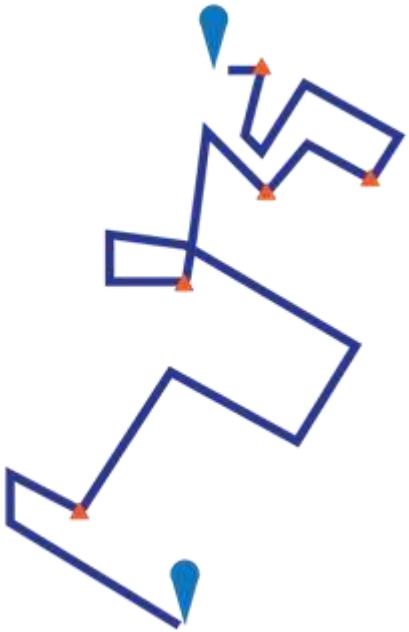
What kind of user authentication?

REST vs SOAP?

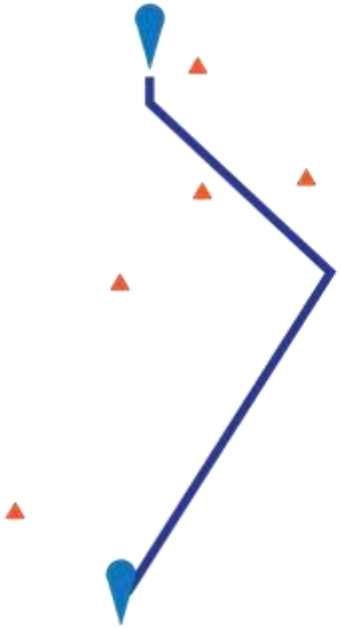
How to architect for scalability?

Uninformed Decisions Lead Your Project Down Suboptimal Paths

Developers



Dev + Ops



Takeaways

People

Digital transformation requires heavy collaboration between all stakeholders

- Continuous design / architecture decisions
- Agreed-on environment / network configuration
- Continuous secure deployment planning
- Continuous secure code review

Digital transformation requires constantly available, open communication channels:

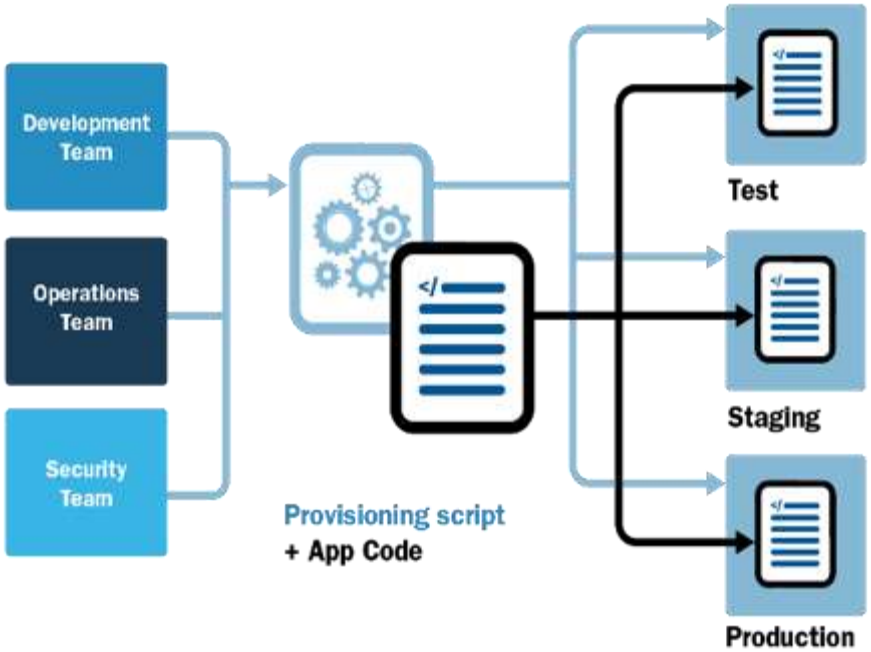
- All business unit should work together in all project decision meetings, virtually or physical but sharing a common collaboration environment
- Digital offerings require a fast cycle of decisions and actions across functional and business line silos.

Process and Practices

Establish a *process* to enable *people* to succeed using the *platform* to develop "right" applications such that:

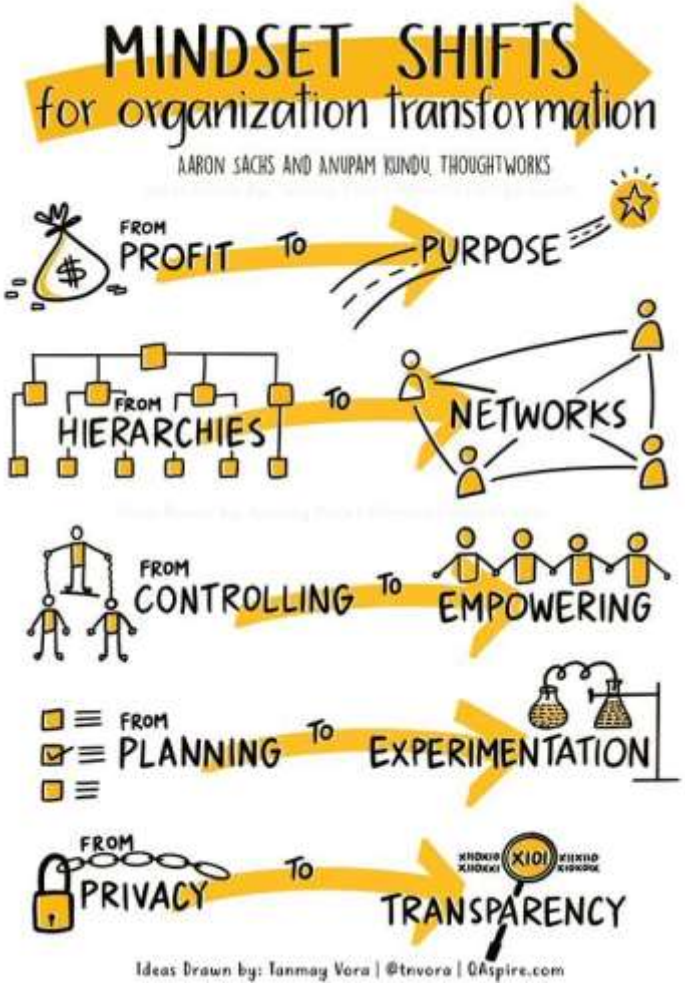
- communication is constant and visible to all
- tasks are testable and repeatable
- human experts are free to do challenging, creative work
- tasks can be performed with minimal effort or cost
- teams have confidence in task success after past repetitions
- deployment is faster, and quality releases are more frequent
- A robust operational backbone is necessary
- Less Bureaucracy , more empowering individual

Platform

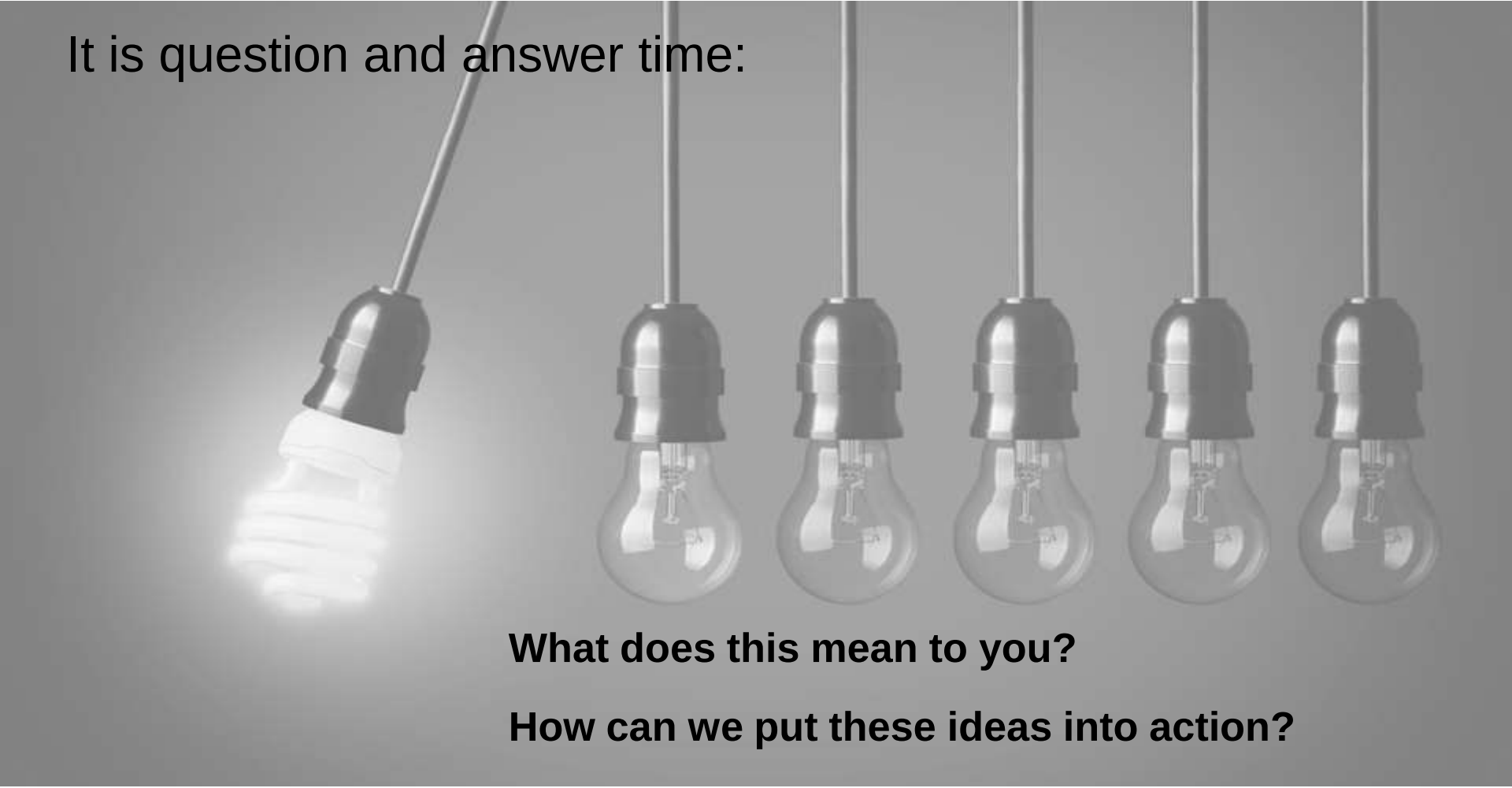


Where **people** use **process** to build software:

- Create a digital platform, not an isolated apps.
- Automated environment creation and provisioning
- Sharing and versioning of environmental configurations
- Collaborative environment between all stakeholders
- Enable common data model & collection platform



It is question and answer time:



What does this mean to you?

How can we put these ideas into action?

DSOI team GitHub Projects

- Once Click DevOps deployment
  https://github.com/DSOI-ALL/devops-microcosm
- Sample app with DevOps Process
  https://github.com/DSOI-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

For more information…

DevOps: https://www.sei.cmu.edu/go/devops

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

Webinar : https://www.sei.cmu.edu/publications/webinars/index.cfm

Podcast : https://www.sei.cmu.edu/publications/podcasts/index.cfm

Thank you

Hasan Yasar
Technical Director
Continuous Deployment of Capability
hyasar@cmu.edu