Hack the Machine
DevOps on the Edge

September 2018

Timothy A. Chick

tchick@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

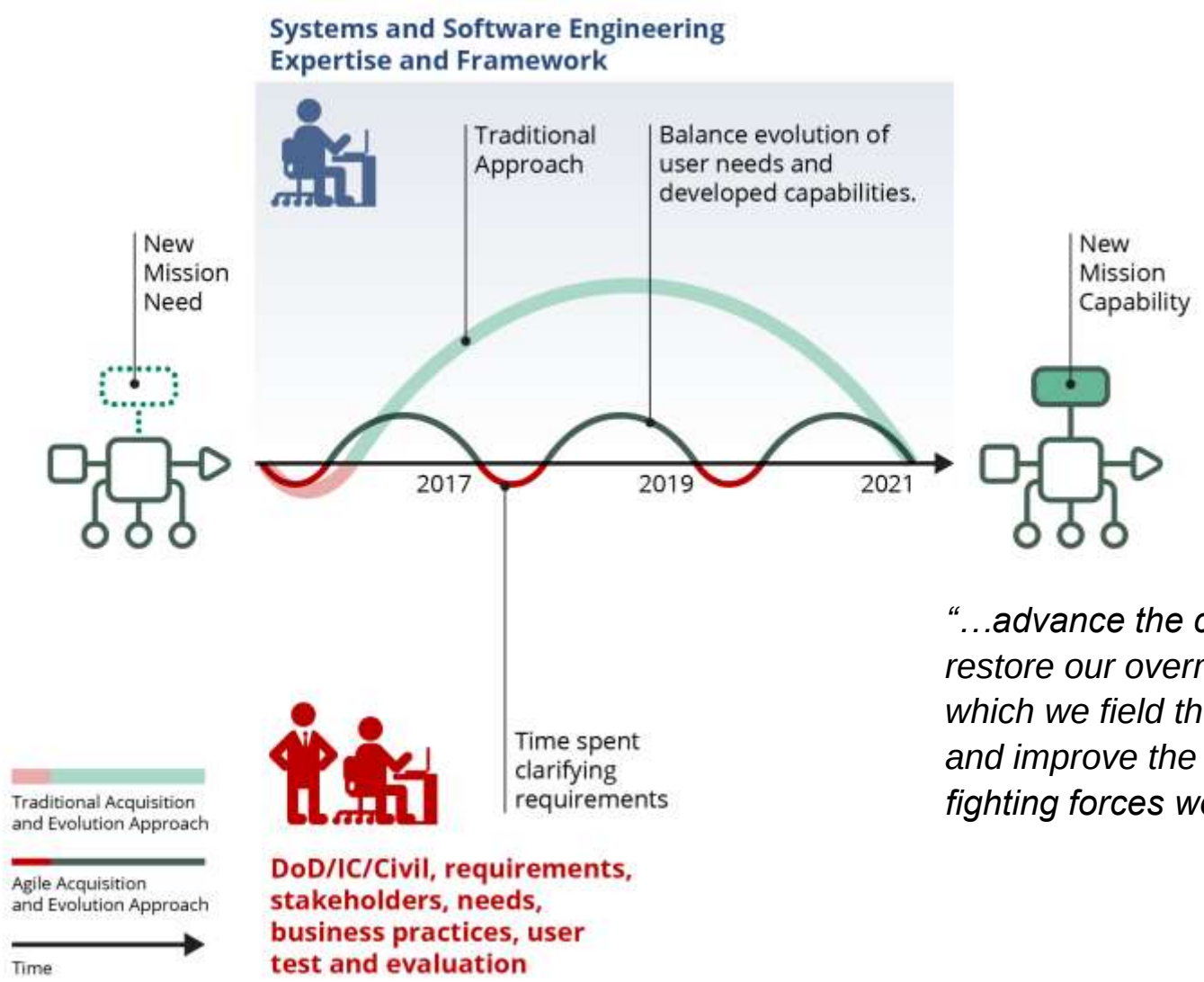
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-1094

Delivering Performance at the Speed of relevance

Streamline rapid, iterative approaches from development to fielding

National Defense Strategy Summary Jan 2018



"…advance the capabilities required to restore our overmatch, speed the rate in which we field these advanced capabilities, and improve the overall affordability of our fighting forces weapon systems…"

Honorable Ellen Lord
Under Secretary of Defense for Acquisition and Sustainment

Large Software Projects Rarely Succeed

Project Size	Successful*
Grand	6%
Large	11%
Medium	12%
Moderate	24%
Small	61%

Source: Standish Group 2015 CHAOS Report

Advantages of small, incremental deliveries

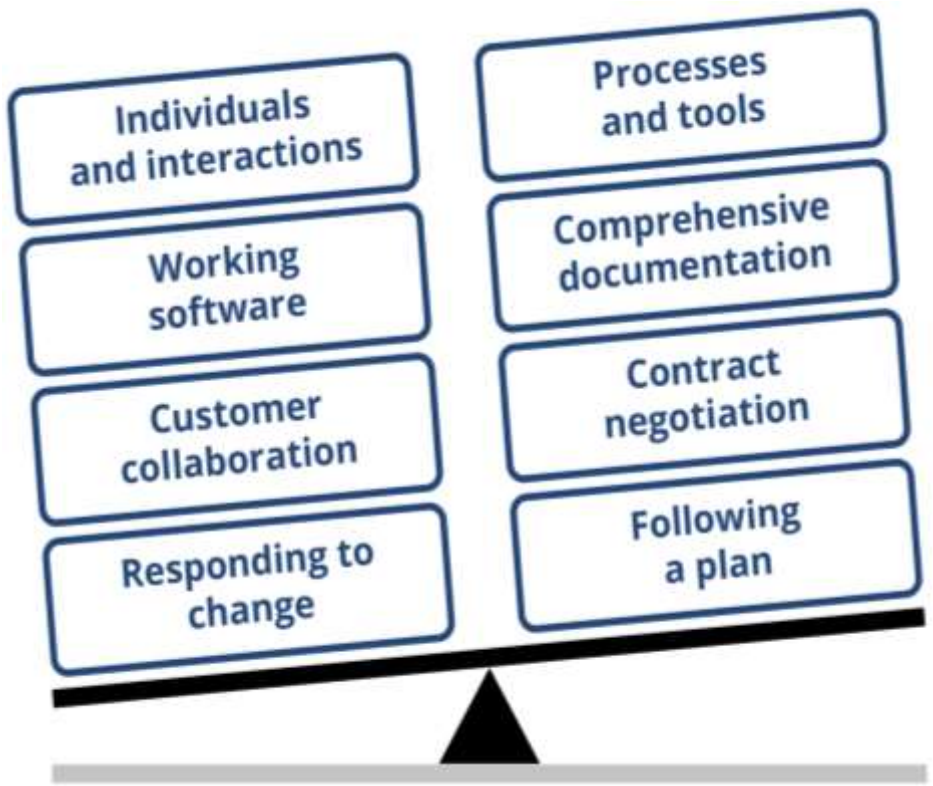
- Fast feedback from stakeholders
- Less investment to move project goals forward
- Less time spent refining low priority items

* **Success:** On Time, On Budget, Satisfactory Result

MITRE Software Engineering Institute

Agile Manifesto

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.

Common myth:

The manifesto is often misinterpreted to mean:

no documentation, no process, and no plan!

http://www.agilemanifesto.org/

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…a constant 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

SAFe Lean-Agile principles

#1 - Take an economic view

#2 - Apply systems thinking

#3 - Assume variability; preserve options

#4 - Build incrementally with fast, integrated learning cycles

#5 - Base milestones on objective evaluation of working systems

#6 - Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7 - Apply cadence, synchronize with cross-domain planning

#8 - Unlock the intrinsic motivation of knowledge workers

#9 - Decentralize decision-making

DevOps aims to Increase…

…the pace of **innovation**

…**responsiveness** to business needs

…**collaboration**

…software **stability and quality**

... continuous feedback

DevOps is an Extension of Agile Thinking

Agile

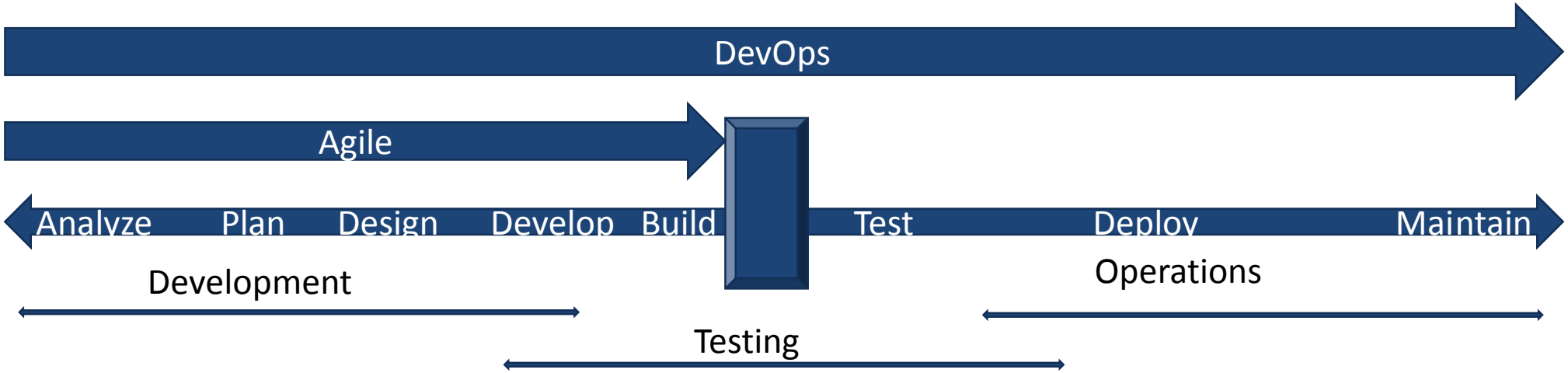
Embrace constant change

Embed Customer in team to internalize expertise on requirements and domain

DevOps

Embrace constant testing, delivery

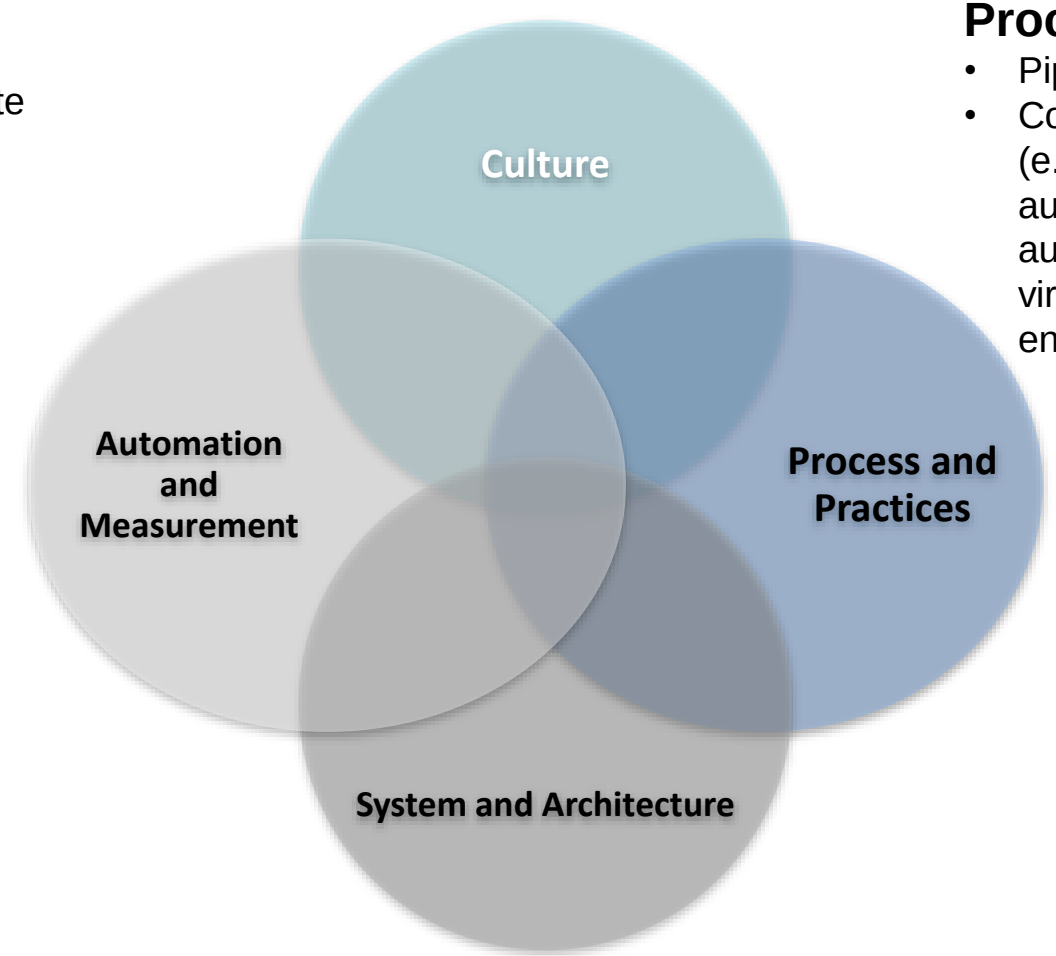
Embed Operations in team to internalize expertise on deployment and maintenance



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.

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

Competing incentives make matters worse on collaboration

- **Development** teams are rewarded for **change**
- **IT Operations** teams are rewarded for **stability**
- **Security** teams are rewarded for tight **security**

Barriers to Adopting DevOps in DoD . . . But DevOps Helps

- Legacy technology, ***re-architect, enable dynamic integration of systems***
- Complex and dependent systems of systems(H/W), ***multiple CI lines***
- Very high risk on continues testing (fail to learn), ***test automation with SimMod***
- Long and paper driven RMF / ATO/ Accreditation process, ***early planning and continuous monitoring, CI***
- Air gapped environment, ***environment parity, IaC, Deployment container***
- Acquisition Process (Contractors, Vendor-COTS, external development), ***agile mindset, hard requirements to capability, early porotype development/integration***
- And Bureaucracy with many pages documentation, ***continuous and integrated documentation as code***

DevOps: Summary: IaC, CI, CD

- Collaboration, Automation, Integration, continuous and early feedback
- Tailor the DevOps principles for the program needs
- Take the advantage of IaC, Deployment, and Test automation
- On new architect, use micro-services architecture and keep as self contained
- Treat compliance as code and integrated to DevOps Development Pipeline
- Integrate security throughout the development pipeline and have continuous security
- Lastly . . .
  - *disciplines of software development with supported infrastructure, new forms of automation and continuous feedback enable faster delivery of the product*