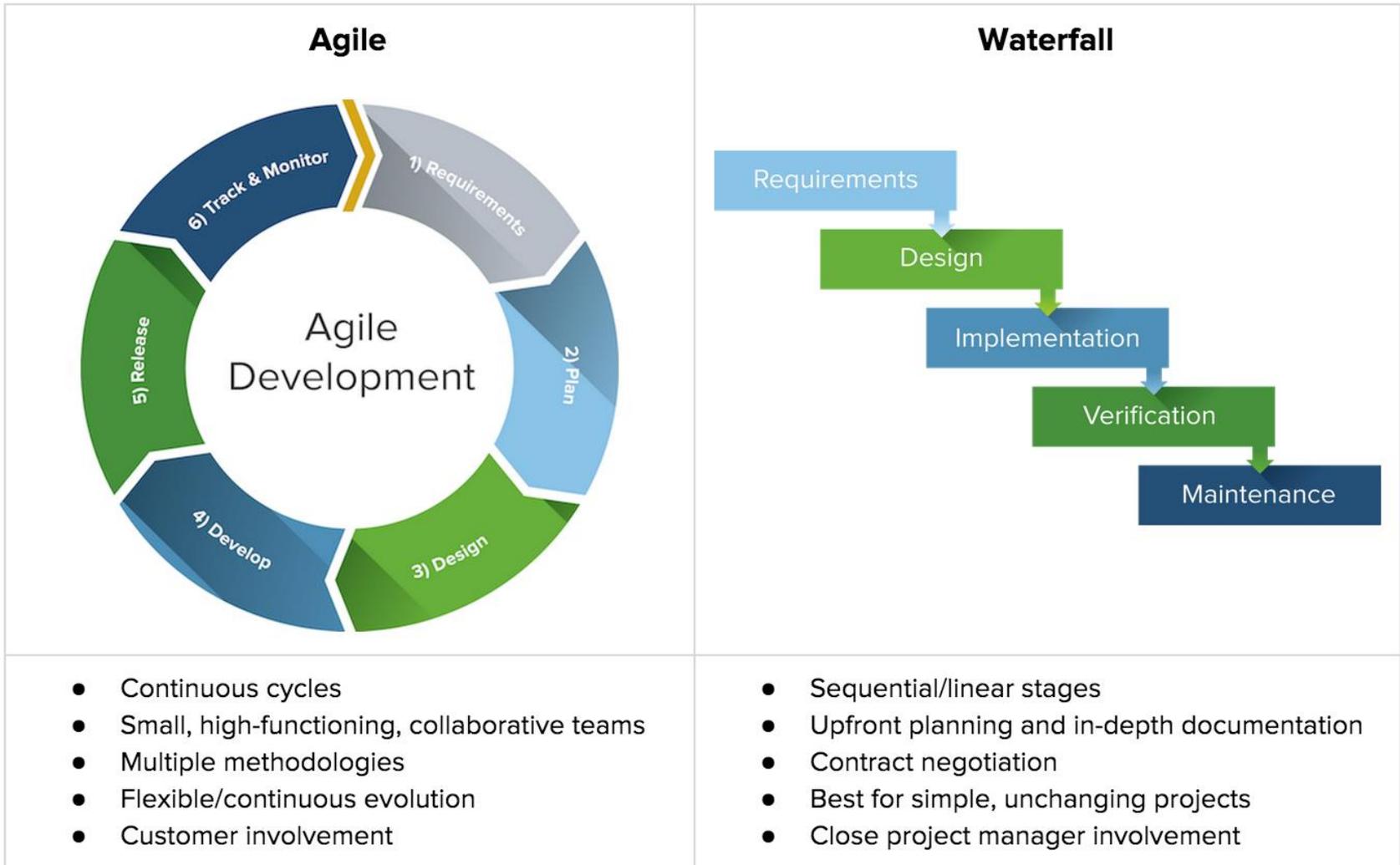


Two-Speed Architecture Governance

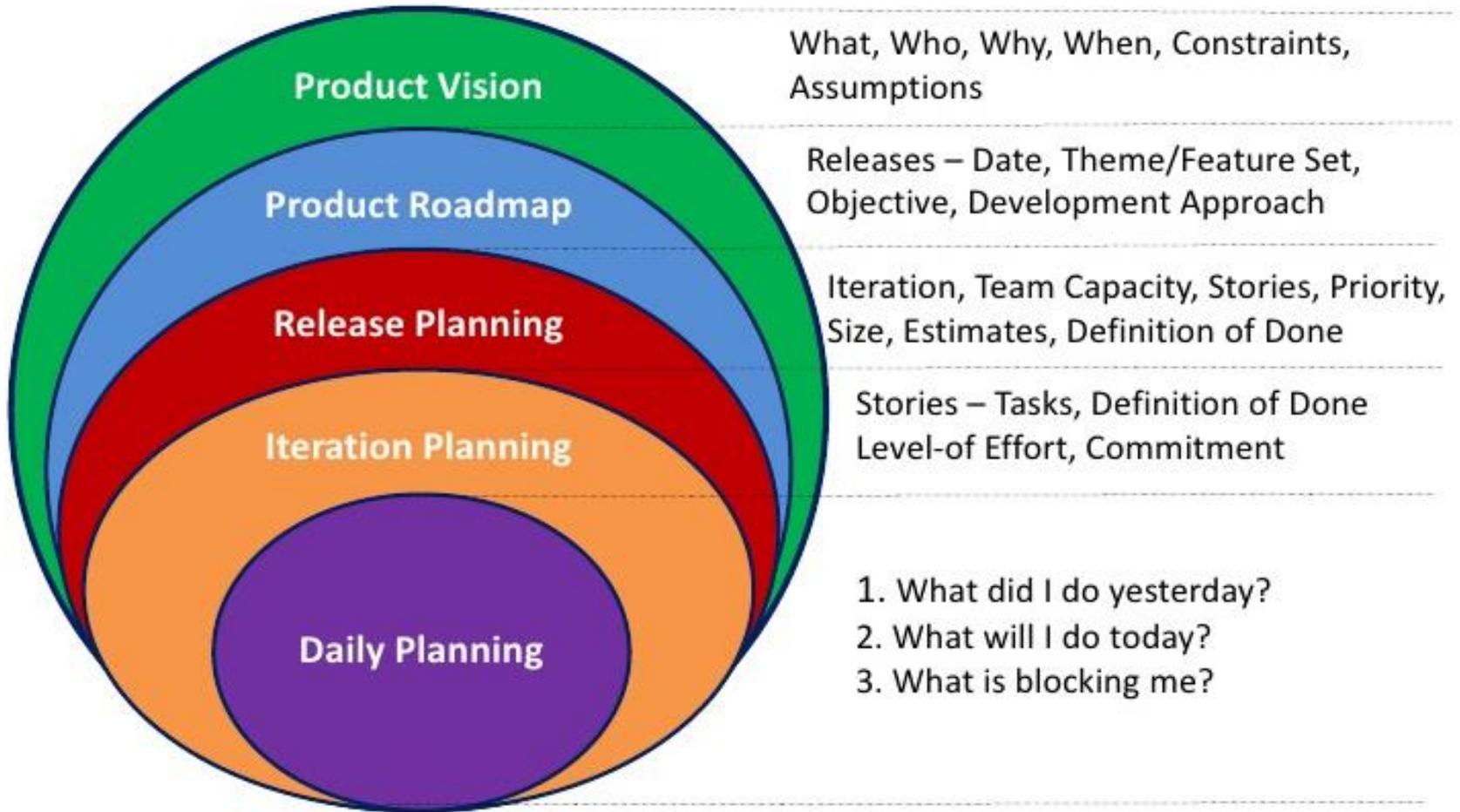
Key Learning Objectives

- Keep these two broad industry changes top of mind, “software has eaten the modern world,” and “RESTful API’s will consume the rest.”
- Agile began with an manifesto, followed with methodology, then planning guidance – governance is the last step in the evolution – and typically the slowest.
- A correct balance between team oversight and the right level of trust, increases the rarity of severe outbreaks of technical debt or rework.
- As far as working software is a concern, software does not get cheaper if you wait.
- The waterfall is not dead, and it will no die any time soon.

Agile versus Waterfall



Agile Planning 🗝️



The nature of the conflict



SPEED OF OPERATIONS

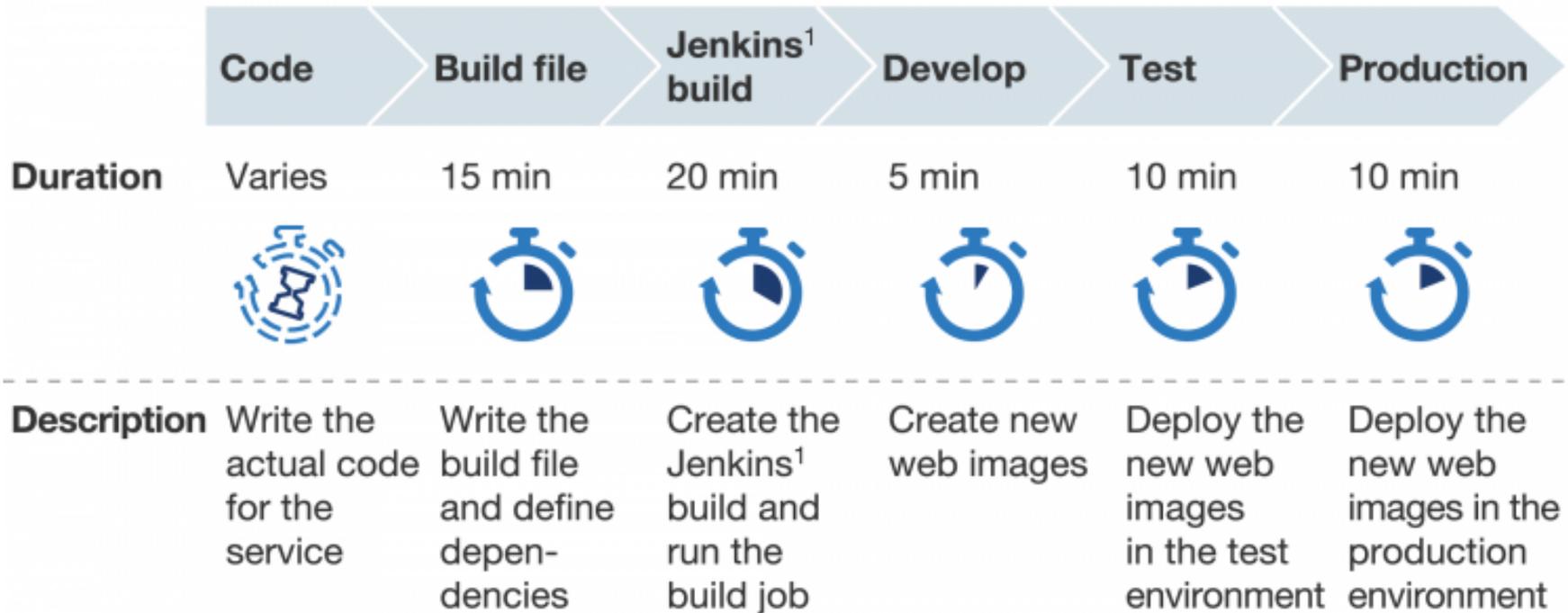
Run stable systems with new approaches



SPEED OF INNOVATION

Agile, fast, just-good enough techniques to explore and adapt to new opportunities

It is possible to deploy new code on a site within an hour.

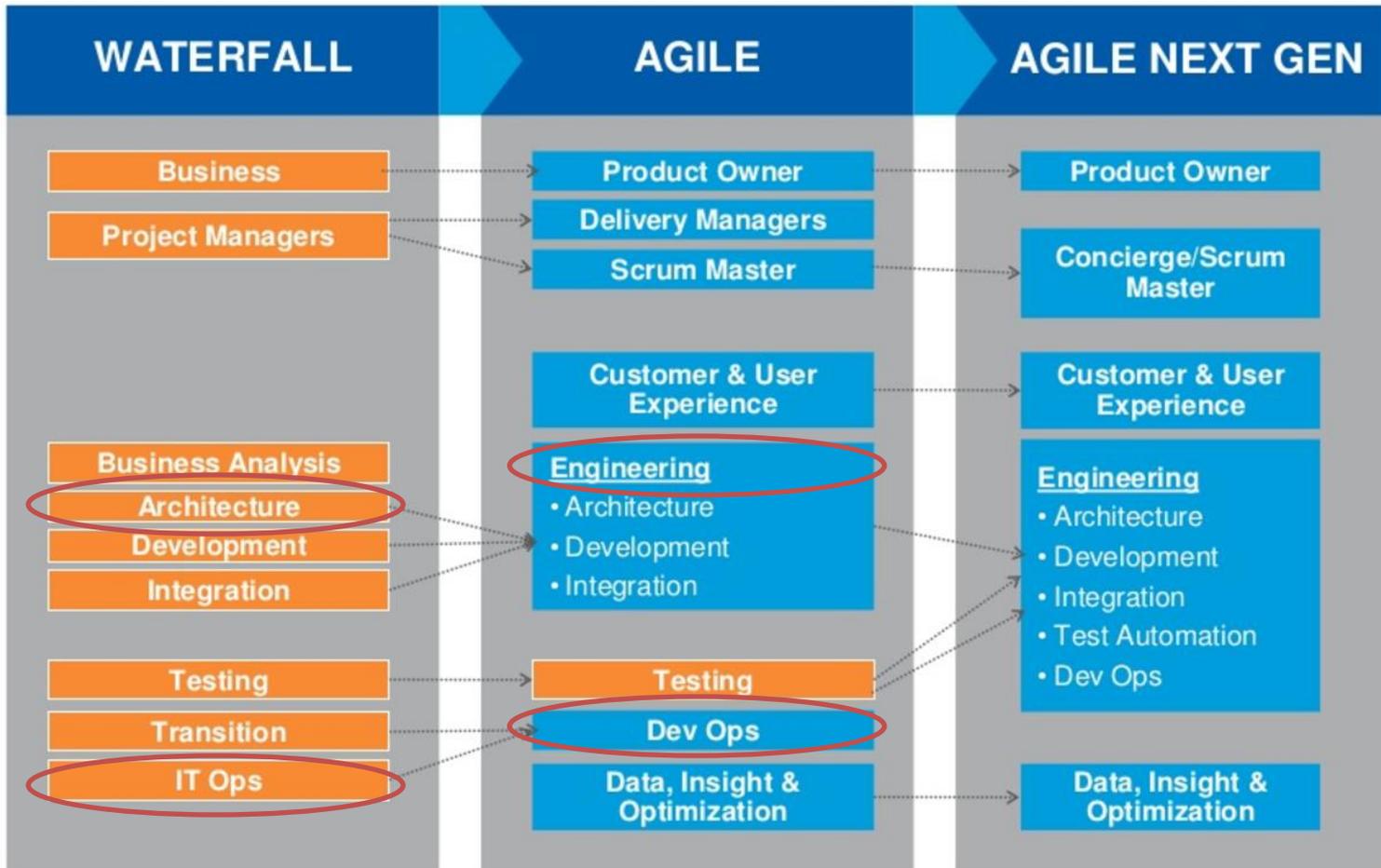


¹Jenkins is an open-source continuous-integration application that monitors execution of repeated jobs, such as building a software project.

McKinsey&Company

How Roles are Changing

Low automation → High automation



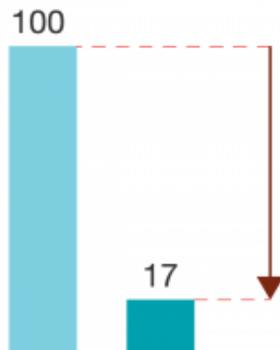
The value of adopting DevOps can be significant.

Indexed to 100

■ Pretransformation ■ Posttransformation

Improvement in time to market

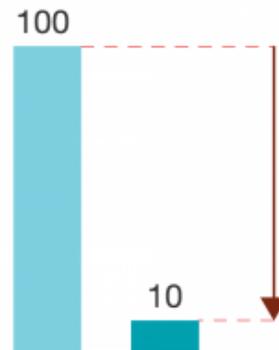
Average number of days from code completion to live production



- Eliminate rework through integrated change management and automated deployment and testing

Reduction in cycle time

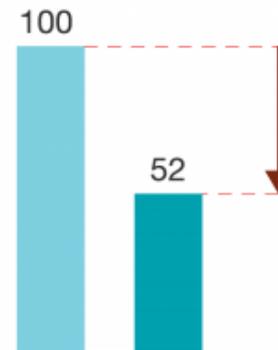
Number of days to update servers and the IT environment



- Eliminate wait time and rework through standardized processes
- Eliminate non-value-added work through automation

Improvement in productivity

Average number of DevOps handoffs per processing activity



- Eliminate wait time and rework through improved development and operations communication

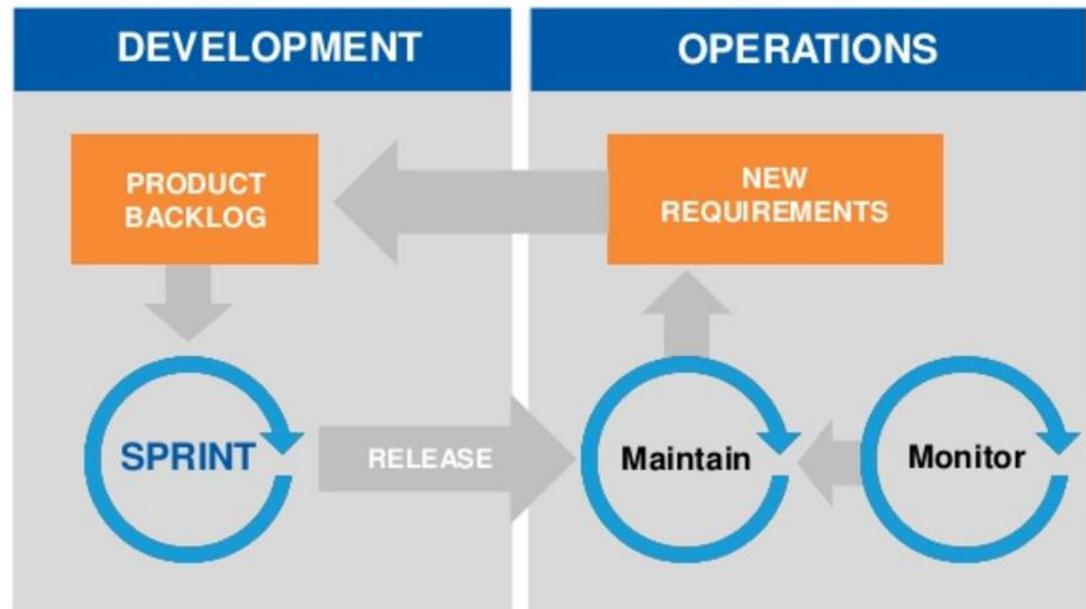
McKinsey&Company

Where is the pressure coming from?

Monolithic IT operations simply move too slow in the modern multi-release market. In response, many lines of business who can control their own IT dollars choose to invest in cloud-based products, or change over development teams to AGILE.

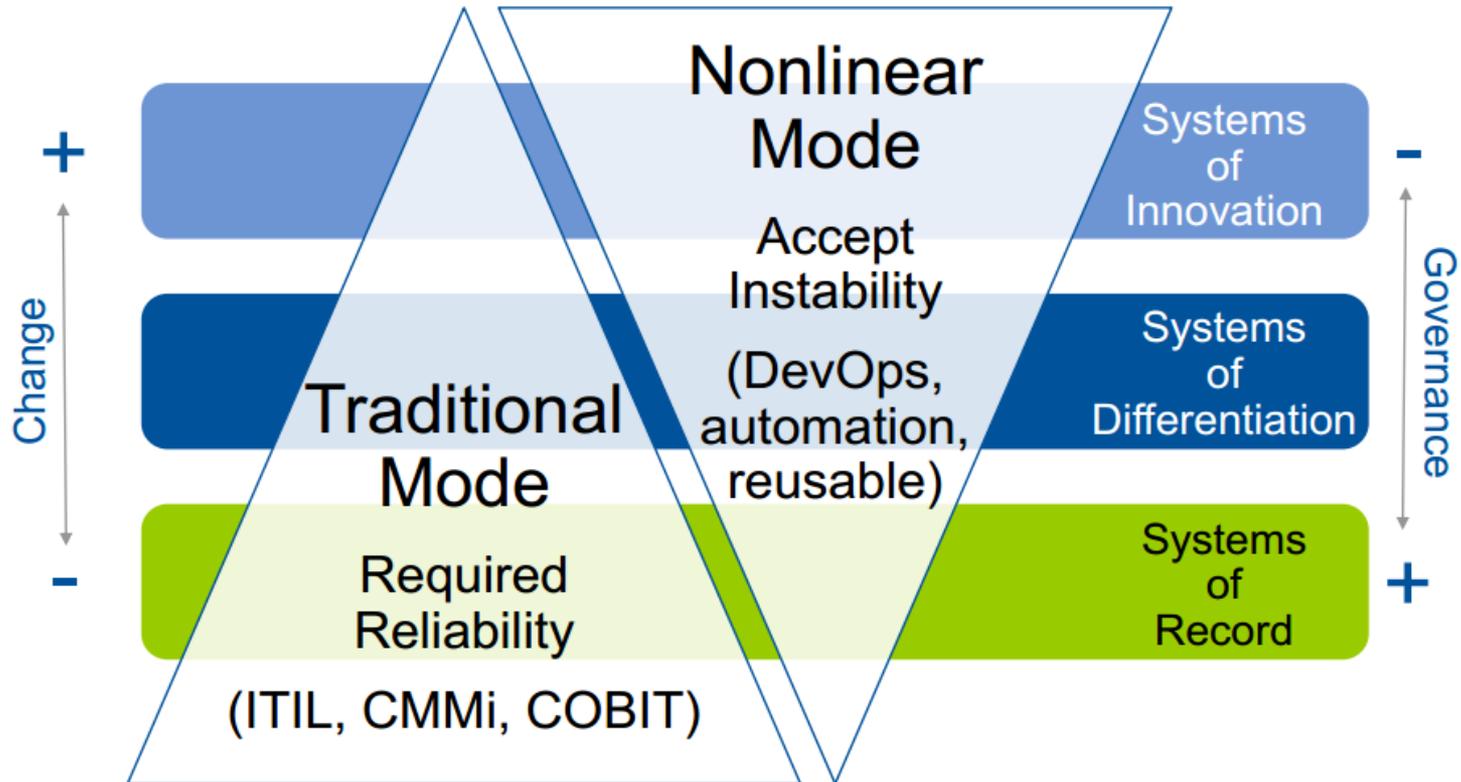
In turn, the pressure on IT is to segment into **CHANGE, BUILD, and RUN/OPERATE** organizations with process automation, analytics, and **DEVOPS** as the driver.

Organizational 2-Speed Nirvana



Achieved by consuming API's

Type-1 vs. Type-2 Governance



© 2014 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner

The meltdown commences ...

Presently, EA manages from a 1-speed frame of reference and not addressing a fundamental question: Which systems and processes within the overall architecture need to move faster?

Presently, IT traditionally has operated via lengthy, centralized planning processes and governance that are often far removed from the business requirements.

The need: Address the technology architecture and infrastructure required to enable DEVOPSs while simultaneously making changes to various operations, processes, and governance structures in the IT organization and within the business overall.

Governance vs. Manifesto

Agile Governance needed

- **PRO** - At the project level identify and advocate for specific goals to be reached during each sprint or iteration to determine the core requirements that comprise the initial release of the project.
- **PRO** - At the program level, the more comprehensive strategic goals of the organization filter down to individual Agile projects that produce tangible results.

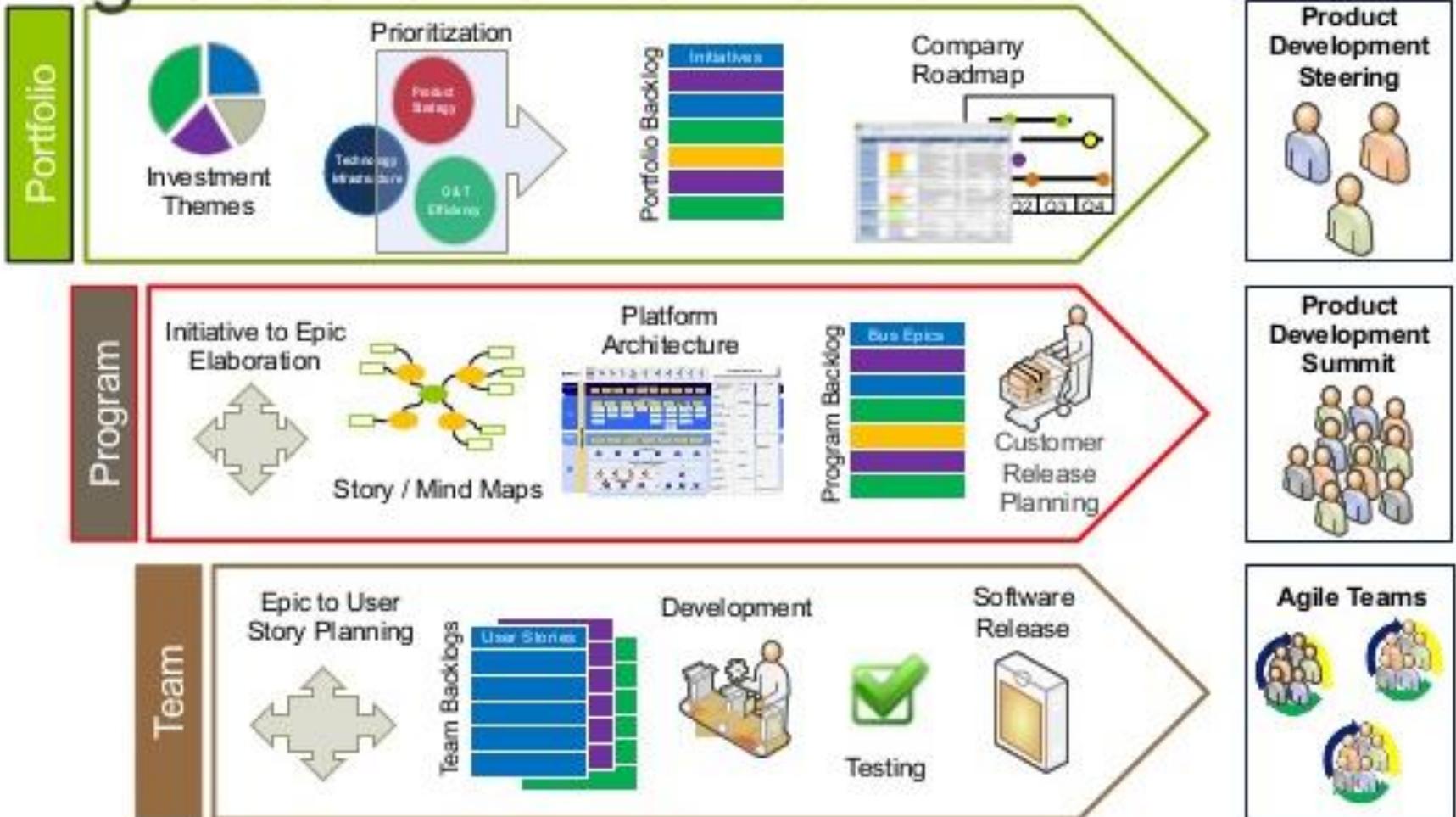
Agile Principles Conflict

- **CON** - Governance requires all kinds of documentation that agile methodologies might deem redundant or have no intrinsic value
- **CON** - Governance and compliance is normally satisfied by upfront analysis and agreement on outputs, which is in conflict with agile principles

Main Agile Risks

- Iterative Risk - As with any design that is not fully completed at the time the first build task is started, there always remains the risk that some outputs may not be fully understood until reached.
- Compliance Risk - A common theme amongst dealing with governance and compliance requirements is that the disciplines required by the agile project management team will satisfy and create the outputs necessary for compliance
- Competence Risk - Not having sufficient agile competencies or training within a project team is a risk in of itself, and, is the leading cause of agile project failure.
- Documentation Risk - The agile process and architecture are sometimes mistakenly thought of as free form processes in which no documentation is required.
- Deceleration Risk – the Agile project will experience push back from the various serial and waterfall-oriented committees responsible for controlling software development within the organization.

Agile Governance Framework



Hard Problems in Agile Governance

- Deciding what issue must elevate out of Type-2 workflow (Agile/DEVOPS) to Type-1 governance for deliberation (e.g. value judgement, alignment, resources, risk and performance management)?
- How to distinguish the difference between oversight, and *pretending* to follow the rules for the sake of the team
- The definition of “done” cannot meet the requirements of TOGAF Phase H Architecture Change Management. The increments are too small.

Essential Questions

- Does EA, from a TOGAF point of view, have a place in Agile Methods?
- Does Agile methods, have a place in TOGAF EA methods?
- Can an enterprise architect function as an EA, in an Agile Program? Is there any work for an EA to perform, or is it all solutions-like work?