# OTHER PHILOSOPHIES

Section 7a

# AGENDA



## OTHER PHILOSOPHIES

- Lean
- Agile
- DevOps
- COBIT

## Other philosophies

ITIL Guiding Principles are reflected in many other frameworks, methods, standards, philosophies, and/or bodies of knowledge, such as **Lean**, **Agile**, **DevOps**, and **COBIT**

This allows organizations to effectively integrate multiple methods into an overall approach to service management

## Lean thinking

is a management framework made up of a philosophy, practices and principles which aim to help practitioners improve efficiency and the quality of work.

Lean thinking encourages whole organisation participation. The goal is to organise human activities to deliver more benefits to society and value to individuals while eliminating waste.

## Fun fact

The term "lean thinking" was coined by mechanical engineer and MIT graduate student John Krafcik in 1988, who subsequently went on to run Google LLC's autonomous driving unit for many years.

Lean thinking is a business methodology adopted in various fields:

- Lean construction, an adaptation of lean manufacturing principles to the design and construction process
- Lean government, application of lean thinking to government
- Lean higher education, application of lean manufacturing principles in Higher Education
- Lean integration, application of lean manufacturing principles to data and systems integration
- Lean IT, application of lean manufacturing principles to the development and management of information technology (IT) products and services

# OTHER PHILOSOPHIES



- Lean laboratory, application of lean manufacturing principles in a laboratory
- Lean manufacturing, a process improvement discipline
- Lean product development, lean thinking applied to product development
- Lean project management, application of lean concepts to project management
- Lean services, application of lean manufacturing principles in a service operation
- Lean software development, lean manufacturing principles applied to software development
- Lean startup, how to start a company in a lean way

## Lean IT

is the extension of lean manufacturing and lean services principles to the development and management of information technology (IT) products and services.

Its central concern, applied in the context of IT, is the elimination of waste, where waste is work that adds no value to a product or service.

Although lean principles are generally well established and have broad applicability, their extension from manufacturing to IT is only just [when?] emerging.

## Types of waste (MUDA) 1/2

| Waste element | Examples | Business outcome |
|---|---|---|
| Defects | •Unauthorized system and application changes.<br>•Substandard project execution. | Poor customer service, increased costs. |
| Overproduction (overprovisioning) | •Unnecessary delivery of low-value applications and services. | Business and IT misalignment, Increased costs and overheads: energy, data center space, maintenance. |
| Waiting | •Slow application response times.<br>•Manual service escalation procedures. | Lost revenue, poor customer service, reduced productivity. |
| Non-Value Added Processing | •Reporting technology metrics to business managers. | Miscommunication. |

## Types of waste (MUDA) 2/2

| Waste element | Examples | Business outcome |
|---|---|---|
| Transportation | •On-site visits to resolve hardware and software issues.<br>•Physical software, security and compliance audits. | Higher capital and operational expenses. |
| Inventory (excess) | •Server sprawl, underutilized hardware.<br>•Multiple repositories to handle risks and control.<br>•Benched application development teams. | Increased costs: data center, energy; lost productivity. |
| Motion (excess) | •Fire-fighting repeat problems within the IT infrastructure and applications. | Lost productivity. |
| Employee knowledge (unused) | •Failing to capture ideas/innovation.<br>•Knowledge and experience retention issues.<br>•Employees spend time on repetitive or mundane tasks. | Talent leakage, low job satisfaction, increased support and maintenance costs. |

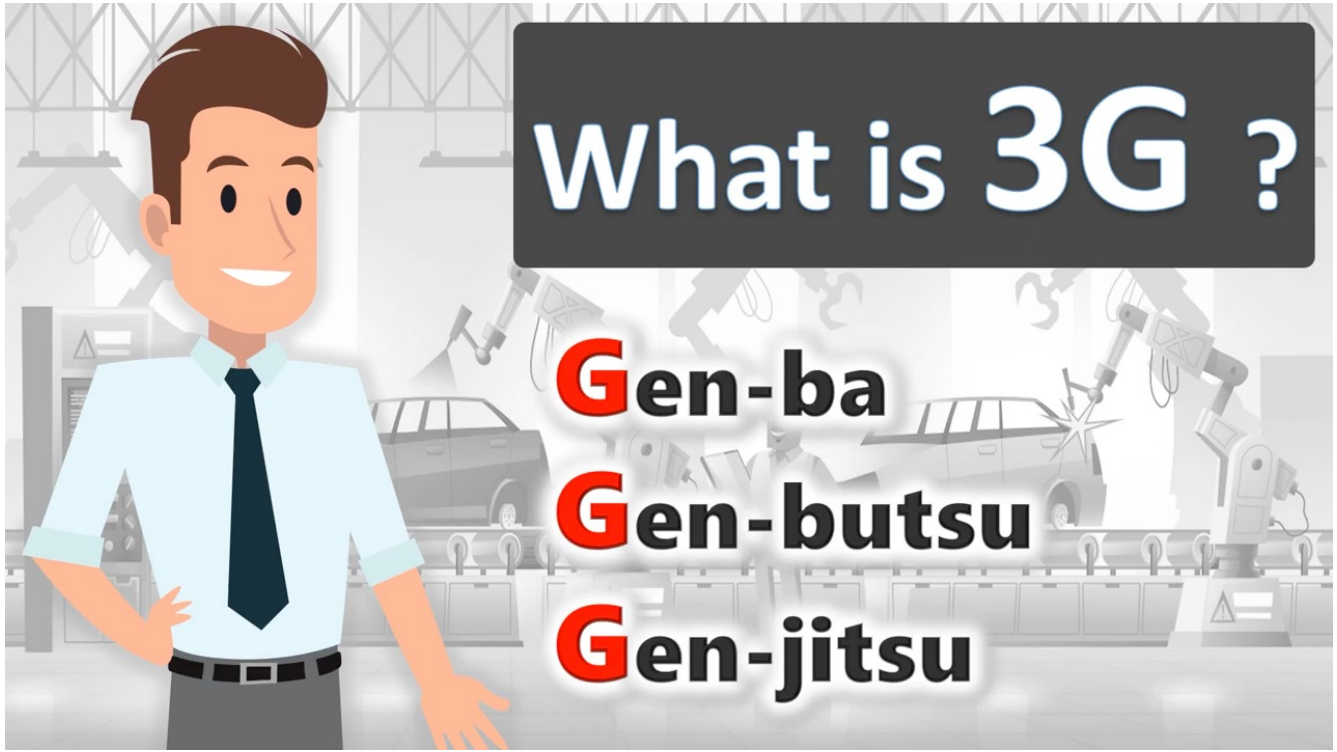
Whereas each element in the table can be a significant source of waste in itself, linkages between elements sometimes create a cascade of waste (the so-called domino effect).

For example, a faulty load balancer (waste element: Defects) that increases web server response time may cause a lengthy wait for users of a web application (waste element: Waiting), resulting in excessive demand on the customer support call center (waste element: Excess Motion) and, potentially, subsequent visits by account representatives to key customers' sites to quell concerns about the service availability (waste element: Transportation).

In the meantime, the company's most likely responses to this problem — for example, introducing additional server capacity and/or redundant load balancing software), and hiring extra customer support agents — may contribute yet more waste elements (Overprovisioning and Excess Inventory).

# OTHER PHILOSOPHIES






https://www.youtube.com/watch?v=xxwYqJ6zpPI

# OTHER PHILOSOPHIES



https://www.youtube.com/watch?v=NFrwxzUhfNA

## Agile

Agile software development is a set of software development methods that originated as a response for the indiscriminate use of CMMI, RUP and PMBOK creating fat and slow software development processes that normally increased the lead time, the work in progress and non-value added/value added activities ratio on projects.

Agile software development methods support a broad range of the software development life cycle.
Some methods focus on the practices (e.g., XP, pragmatic programming, agile modeling), while some focus on managing the flow of work (e.g., Scrum, Kanban).
Some support activities for requirements specification and development (e.g., FDD), while some seek to cover the full development life cycle (e.g., DSDM, RUP).

Some methods focus on the practices (e.g., XP, pragmatic programming, agile modeling), while some focus on managing the flow of work (e.g., Scrum, Kanban).

Some support activities for requirements specification and development (e.g., FDD), while some seek to cover the full development life cycle (e.g., DSDM, RUP).

## Fun fact

Popularized in the 2001 *Manifesto for Agile Software Development*, these values and principles were derived from and underpin a broad range of software development frameworks, including Scrum and Kanban.

# OTHER PHILOSOPHIES



| Framework | Main contributor(s) |
|---|---|
| Adaptive software development (ASD) | Jim Highsmith, Sam Bayer |
| Agile modeling | Scott Ambler, Robert Cecil Martin |
| Agile unified process (AUP) | Scott Ambler |
| Disciplined agile delivery | Scott Ambler |
| Dynamic systems development method (DSDM) | Jennifer Stapleton |
| Extreme programming (XP) | Kent Beck, Robert Cecil Martin |
| Feature-driven development (FDD) | Jeff De Luca |
| Lean software development | Mary Poppendieck, Tom Poppendieck |
| Lean startup | Eric Ries |
| Kanban | Taiichi Ohno |
| Rapid application development (RAD) | James Martin |
| Scrum | Ken Schwaber, Jeff Sutherland |
| Scrumban | |
| Scaled agile framework - SAFe | Scaled Agile, Inc. |

## Agile software development principles

The Manifesto for Agile Software Development is based on twelve principles:

- Customer satisfaction by early and continuous delivery of valuable software.
- Welcome changing requirements, even in late development.
- Deliver working software frequently (weeks rather than months).
- Close, daily cooperation between business people and developers.
- Projects are built around motivated individuals, who should be trusted.
- Face-to-face conversation is the best form of communication (co-location).

## Agile software development principles

- Working software is the primary measure of progress.
- Sustainable development, able to maintain a constant pace.
- Continuous attention to technical excellence and good design.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- Best architectures, requirements, and designs emerge from self-organizing teams.
- Regularly, the team reflects on how to become more effective, and adjusts accordingly.

## Scrum

Scrum is one of the more well known agile methods for project management, and has as one of its origins concepts from Lean Thinking.

Scrum also organizes work in a cross-functional, multidisciplinary work cell.

It uses some form of kanban system to visualize and limit work in progress, and follows the PDCA cycle, and continuous improvements, that is the base of Lean.

## DevOps

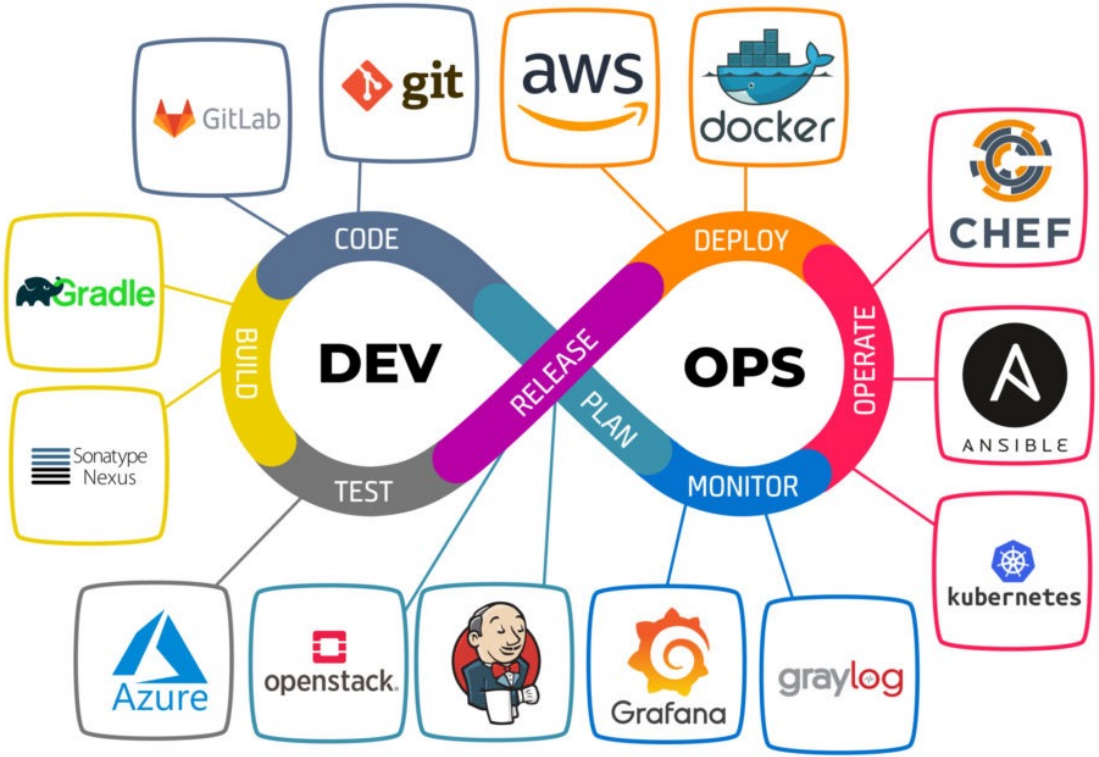
DevOps is a methodology in the software development and IT industry.

Used as a set of practices and tools, DevOps integrates and automates the work of software development (Dev) and IT operations (Ops) as a means for improving and shortening the systems development life cycle.

DevOps is complementary to agile software development; several DevOps aspects came from the agile way of working.

# OTHER PHILOSOPHIES





# OTHER PHILOSOPHIES



https://www.youtube.com/watch?v=kBV8gPVZNEE

# OTHER PHILOSOPHIES



The Seven Guiding Principles are:

- Focus on value
- Start where you are
- Progress iteratively with feedback
- Collaborate and promote visibility
- Think and work holistically
- Keep it simple and practical
- Optimize and automate