Software Engineering Practices

Introduction to Software Systems

Spring 2025

Software Engineering Research Center

IIIT Hyderabad, India

- Software Offerings
- Software Industry
- Software Engineering Teams
- Software Development Life Cycle
- Software Product Methods
- Continuous Development and Continuous Integration
- DevOps and MLOps
- Software Productivity
- Technical Debt
- Software Development Waste

SaaS

PaaS

IaaS

Hosted Application / Apps

Development tools, database management, business analytics
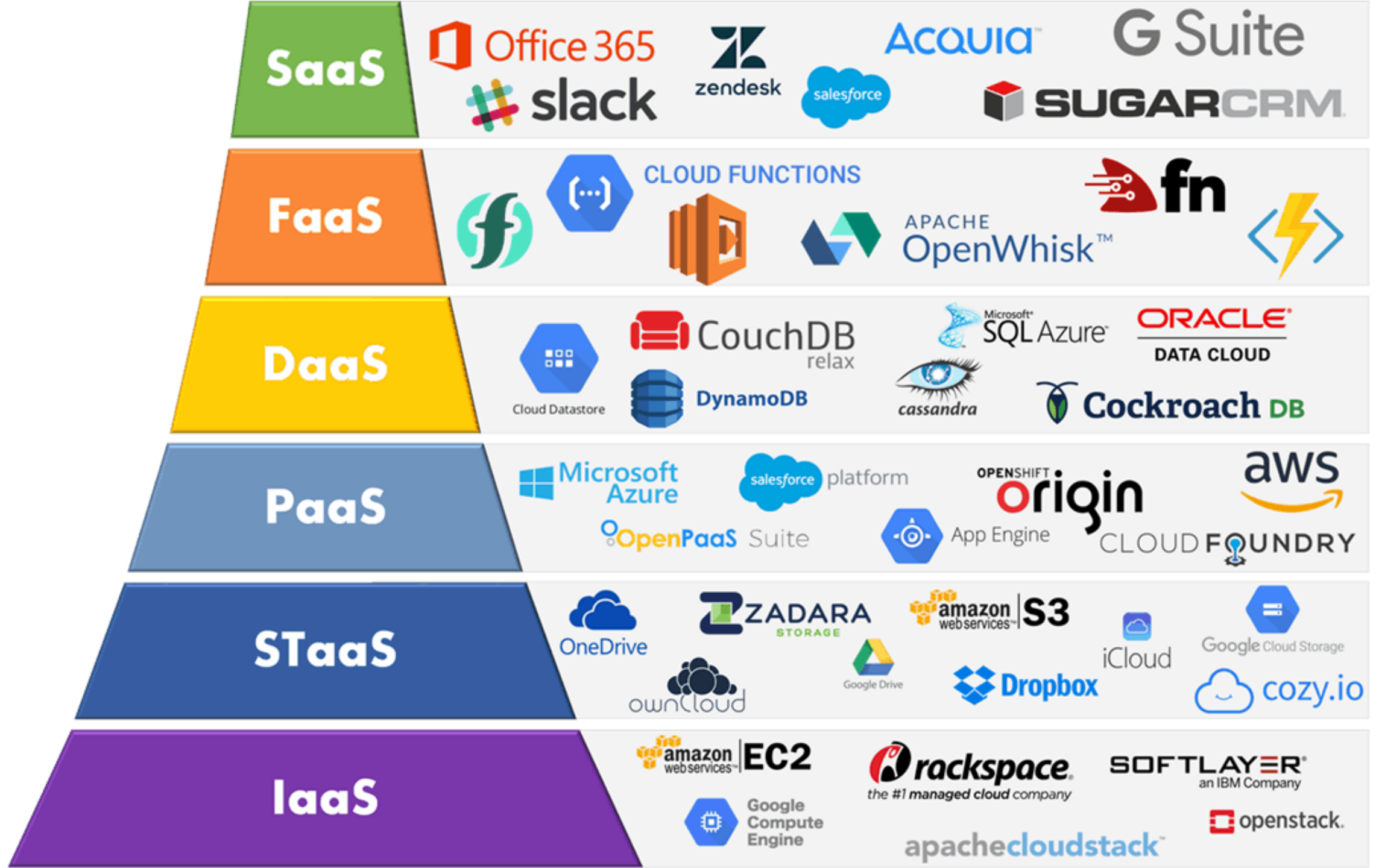
Operating Systems

Servers and storage

Networking firewalls/security

Data center physical plant/building

Cloud computing service models pyramid:

- **SaaS** — Office 365, slack, zendesk, salesforce, Acquia, G Suite, SUGARCRM
- **FaaS** — Fission, Google CLOUD FUNCTIONS, AWS Lambda, Apache OpenWhisk, fn, Azure Functions
- **DaaS** — Cloud Datastore, CouchDB relax, DynamoDB, Microsoft SQL Azure, cassandra, ORACLE DATA CLOUD, Cockroach DB
- **PaaS** — Microsoft Azure, salesforce platform, OpenShift Origin, aws, OpenPaaS Suite, App Engine, CLOUD FOUNDRY
- **STaaS** — OneDrive, ZADARA STORAGE, amazon web services S3, iCloud, Google Cloud Storage, ownCloud, Google Drive, Dropbox, cozy.io
- **IaaS** — amazon web services EC2, rackspace the #1 managed cloud company, SOFTLAYER an IBM Company, Google Compute Engine, apache cloudstack, openstack

Rights granted	Public domain	Permissive FOSS license (e.g. BSD license)	Copyleft FOSS license (e.g. GPL)	Freeware	Proprietary license
Copyright retained	No	Yes	Yes	Yes	Yes
Right to perform	Yes	Yes	Yes	Yes	Yes
Right to display	Yes	Yes	Yes	Yes	Yes
Right to copy	Yes	Yes	Yes	Often	No
Right to modify	Yes	Yes	Yes	No	No
Right to distribute	Yes	Yes, under same license	Yes, under same license	Often	No
Right to sublicense	Yes	Yes	No	No	No
Example software	SQLite, ImageJ	Apache web server, ToyBox	Linux kernel, GIMP, OBS	Irfanview, Winamp, *Legends*	Windows, xSplit, TIDAL DRMs, Spotify,

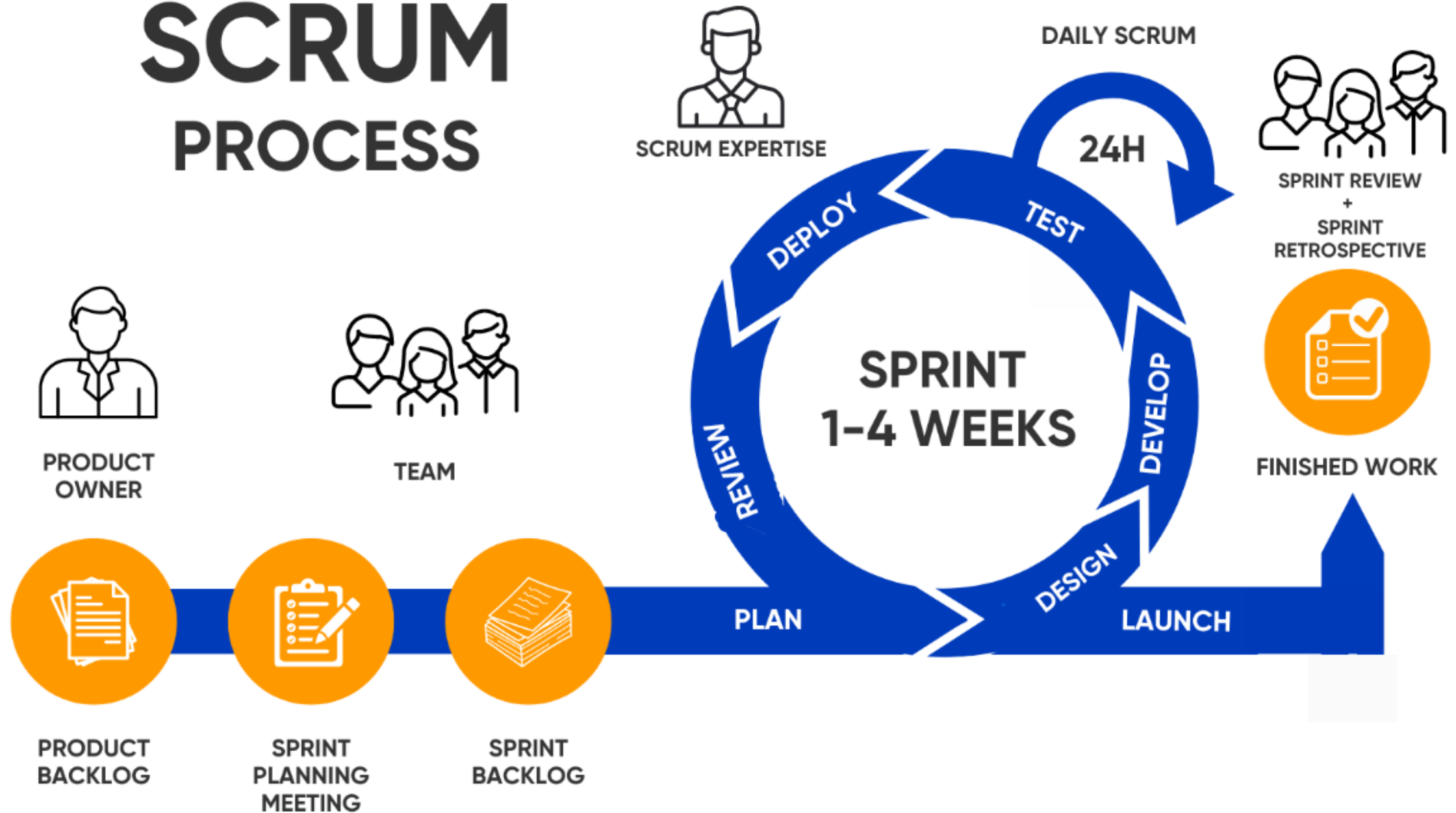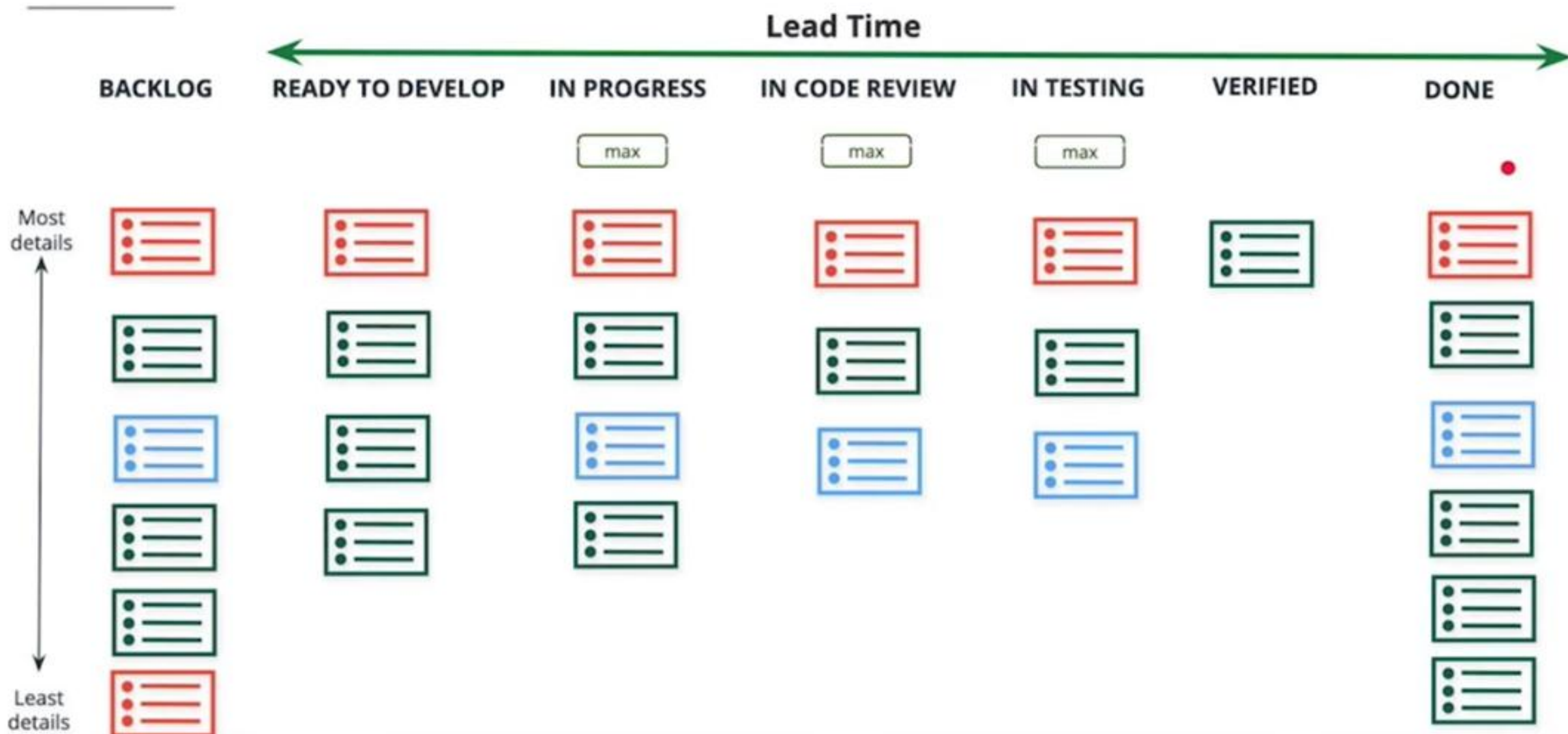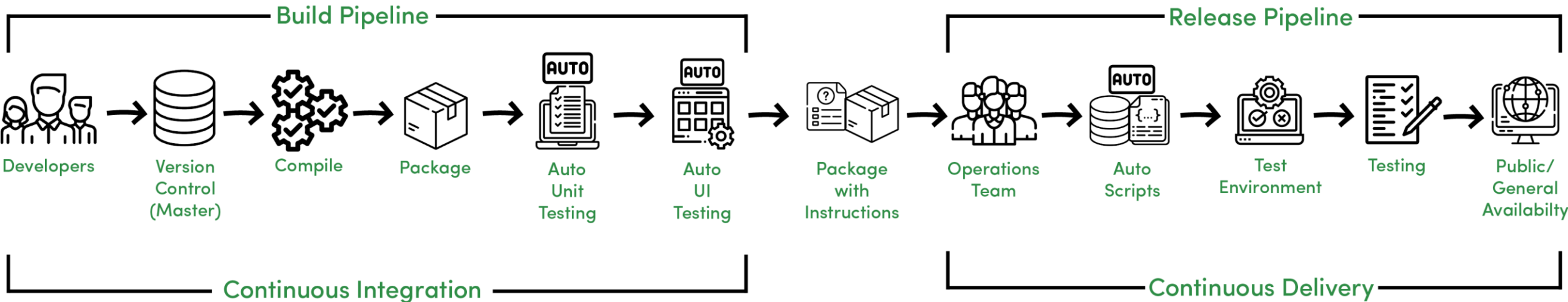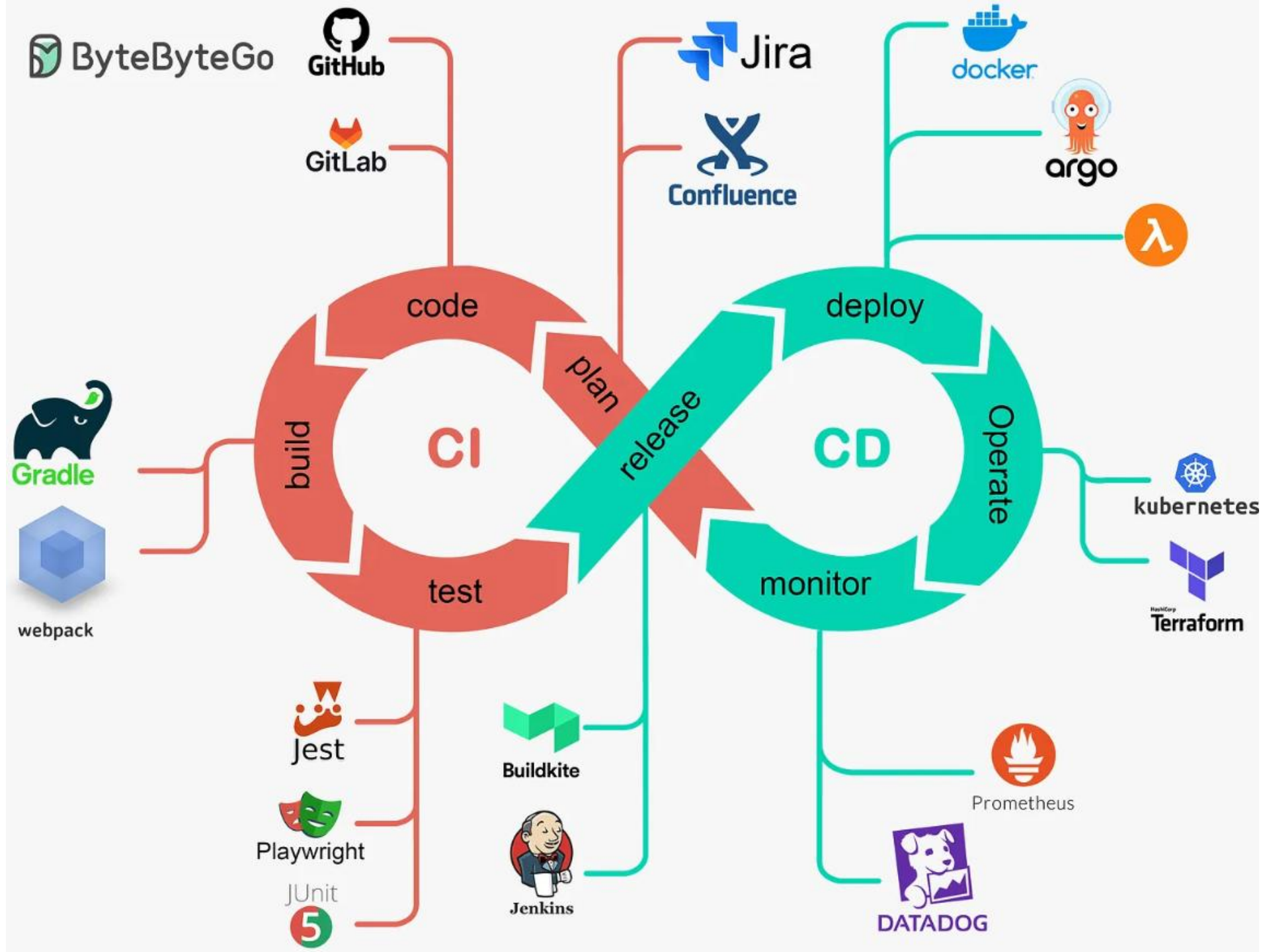|	**© Copyright**	**↺ Copyleft**	**🔑 Permissive**	**CC Creative Commons**
What is a user allowed to do with the code?	What creator dictates	What user wants under certain rules	What user wants with a few restrictions	What user wants without restrictions
Clause of the use	As creator dictates	Derivative work must be attributed to creator, open-source and copyleft	Derivative work must be attributed to a creator	Derivative work must be attributed to a creator
Source code	As creator dictates	Must be open	Don't have to be open	No specific terms about the distribution of source code
Is creator liable for bugs?	✔YES	✔YES	✘ NO	✘ NO
Re-licensing	As creator dictates	Derivative work cannot be released as proprietary software	Derivative work can be released under another license or as proprietary software	Derivative work can be released under another license or as proprietary software
Commercial restrictions	As creator dictates	Permitted	Permitted	Permitted

SCRUM PROCESS

SCRUM EXPERTISE

DAILY SCRUM

24H

SPRINT REVIEW
+
SPRINT RETROSPECTIVE

PRODUCT OWNER

TEAM

DEPLOY

TEST

REVIEW

SPRINT
1-4 WEEKS

DEVELOP

DESIGN

PLAN

LAUNCH

FINISHED WORK

PRODUCT BACKLOG

SPRINT PLANNING MEETING

SPRINT BACKLOG

The Scaled Agile Framework (SAFe) 5.1

Left column (value stream / competencies):
- Organizational Agility
- Lean Portfolio Management
- Agile Product Delivery
- Team and Technical Agility
- Continuous Learning Culture

Top banner:
- Operational Value Streams
- Business Agility
- Measure & Grow

PORTFOLIO
- Enterprise | Government
- Epic Owners, Enterprise Architect
- Strategic Themes, Portfolio Vision
- Kanban — Portfolio Backlog — NFRs
- PB — Lean Budgets — Guardrails
- Epic, Enabler, Epic
- Coordination
- Development Value Streams
- KPIs
- Solutions

ESSENTIAL
- Business Owners
- System Arch/Eng, Product Mgmt, RTE
- Agile Teams
- Product Owner, Scrum Master
- Customer Centricity, Design Thinking
- XP — Plan, Execute, Review, Retro — Scrum
- Kanban
- Kanban — WSJF — Program Backlog — NFRs
- Team Backlogs — NFRs
- Continuous Delivery Pipeline
- Agile Release Train
- Continuous Exploration, Continuous Integration, Continuous Deployment
- Release on Demand
- System Demos
- CD, CI, CE
- Goals, Iterations, PI
- PI Planning, IP Iteration
- Enabler, Feature, Story
- PI Objectives
- Architectural Runway
- Leffingwell, et al. © Scaled Agile, Inc.
- Solution, Solution Context
- Sec, DevOps, Built-In Quality

Right column:
- Vision
- Roadmap
- Milestones
- Shared Services
- CoP
- System Team
- Lean UX
- Metrics

Bottom bar:
- Business | Technology
- Core Values
- Lean-Agile Mindset
- SAFe Principles
- Implementation Roadmap
- SPC
- 5.1
- Lean-Agile Leadership

Build Pipeline

Developers → Version Control (Master) → Compile → Package → Auto Unit Testing → Auto UI Testing →

Continuous Integration

Package with Instructions →

Release Pipeline

Operations Team → Auto Scripts → Test Environment → Testing → Public/ General Availabilty

Continuous Delivery

MLOps= ML + DEV + OPS

ML	DEV	OPS

Experiment	Develop	Operate
Data Acquisition	Modeling+Testing	Continuous Delivery
Business Understanding	Continuous integration	Data Feedback Loop
Initial Modeling	Continuous Deployment	Syatem + Model Monitoring

Data management	Model development	Model deployment	Monitoring andlogging	Model governance and compliance

DataOps CI/CD platform

Feature Store
- Commit-0097
- Commit-0002

Data

Code

MLOps CI/CD platform

git

Data Engineer

git

Data Scientist

Model Training & Model Validation

Model Repo, Serving & Monitoring

DORA 4 key metrics	**SPACE** 5 dimensions	**FLOW METRICS** 5 key metrics
Published in 2018	Published in 2021	Published in 2018
Lead Time for Changes	Satisfaction and Well-being	Flow Velocity
Change Failure Rate	Performance	Flow Efficiency
Deployment Frequency	Activity	Flow Time
Mean Time to Recovery (MTTR)	Communication and Collaboration	Flow Load
	Efficiency and Flow	Flow Distribution
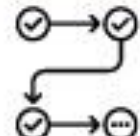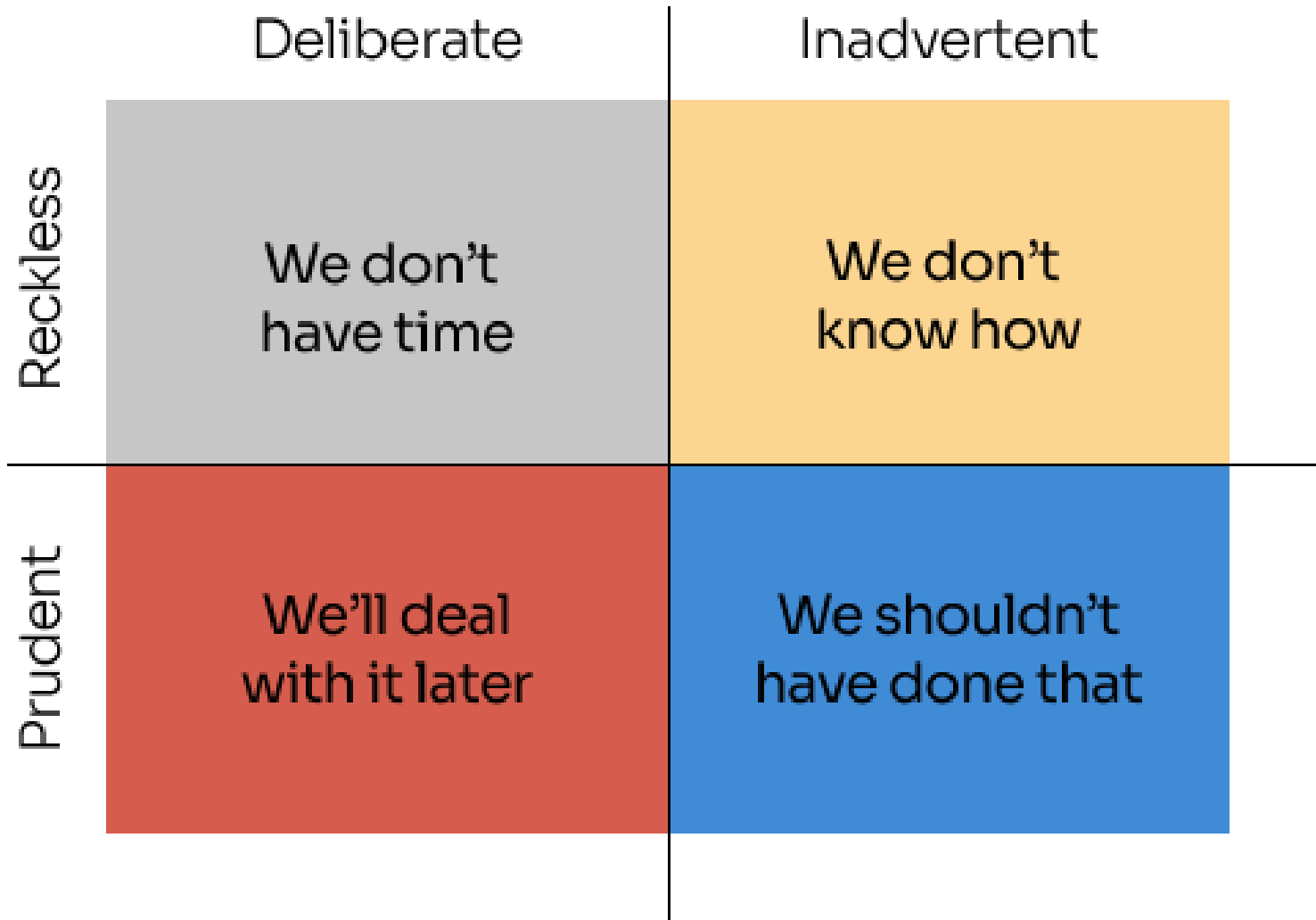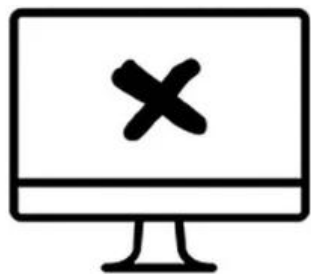
Building the wrong feature or product
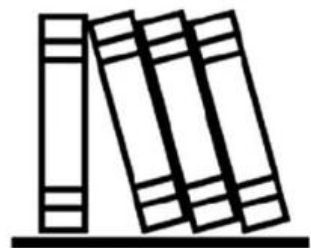
Mismanaging the backlog

Rework
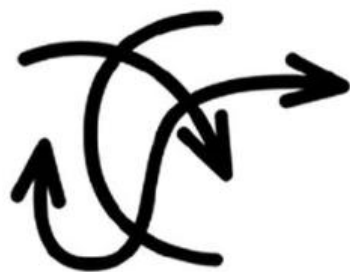
Unnecessarily complex solutions

Extraneous cognitive load

Psychological distress

Knowledge loss

Waiting/multitasking

Ineffective communication