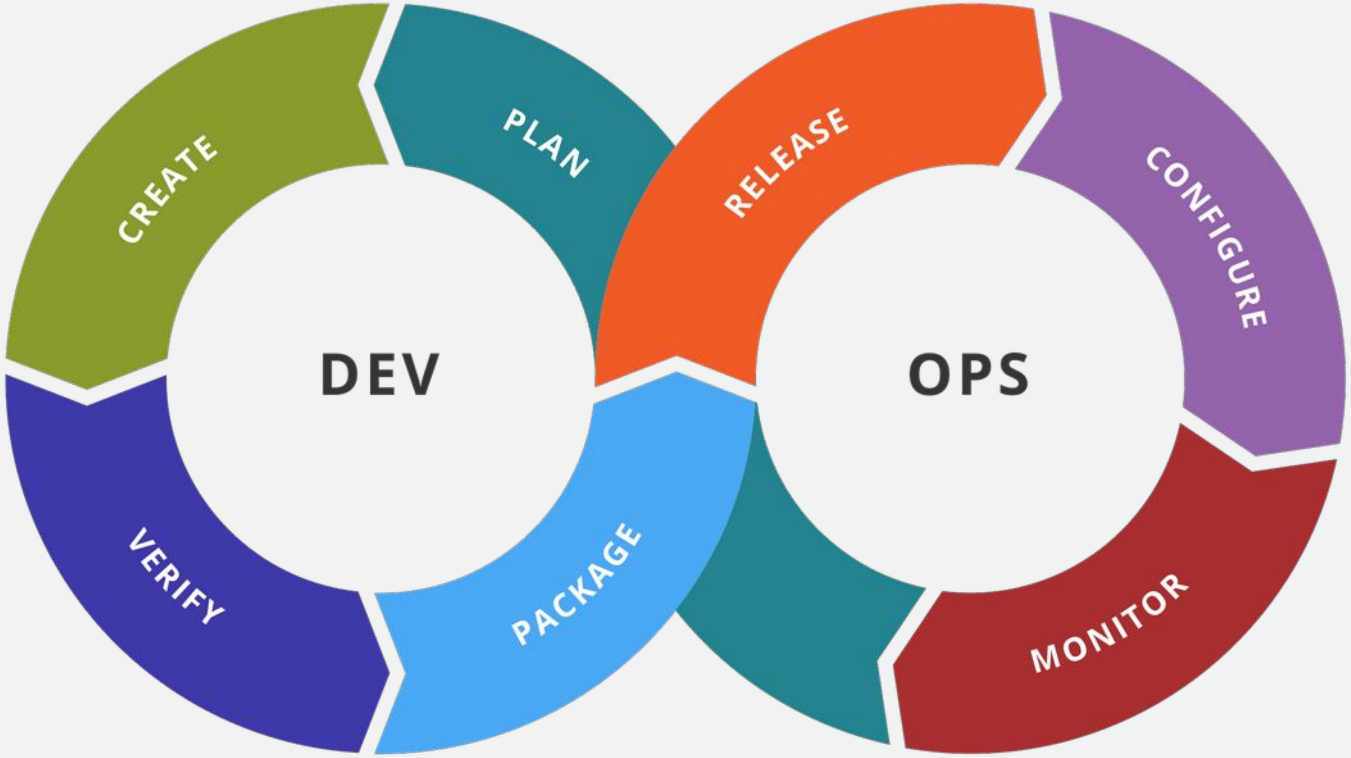
DevOps Mindset and Culture

Alexander Günsche

Senior Solutions Architect
lxg@amazon.com

DevOps Culture



DevOps is a culture of cross-functional collaboration to continuously deliver software product increments.

Digital Products

What is a Digital Product?

A digital product is an intangible offering, primarily software-based or data-driven, that is delivered or accessed through digital channels and technologies. It provides value to users through digital experiences, services, or functionalities.

Product Strategy Canvas

VISION

How can I inspire people to get up every day and come to work?

What are we aspiring to achieve? What values do we uphold?

Start with something simple. Your vision will evolve along with other elements of the strategy.

RELATIVE COSTS

What do we optimize for?

Do we optimize for low cost, like Southwest Airlines, or for unique value, like Starbucks?

Low costs might be a priority, but they do not necessarily mean having low prices.

TRADE-OFFS

Trade-offs define what NOT to do.

IKEA doesn't sell assembled furniture and limits available choices (e.g., materials).

Trade-offs create focus, amplify the value and make the strategy difficult to copy by others without sacrificing their existing businesses.

UNIQUE ACTIVITIES

Define a set of distinct activities in creating, producing, marketing, and delivering your product.

For IKEA: flat packs, warehouses attached to the stores, in-store restaurants, delivery outsourced to the customers, etc.

CAPABILITIES

What competencies and resources do we need to acquire? Do we need suppliers?

Are there any systems necessary to support our strategic choices?

What resources will we need to communicate our strategy effectively?

MARKET

The market is defined by the problems people have. For example, IKEA’s market: people that want to get high-quality home furnishings at low prices.

What are the customer's problems (needs, jobs) worth solving (low satisfaction, high importance)? Within the market, there are groups of people with similar, more specific needs/jobs, goals, and success metrics.

Why do we want to compete in this market, not others? Have we analyzed Porter’s 5 forces or performed PESTEL analysis? Do we know TAM, SAM, SOM, Average Annual Growth Rate, ARPU, Average CAC, and Average Churn Rate?

Are there any constraints, e.g., geography, language?

VALUE PROPOSITION

What key customer needs/jobs do we want to solve?

Which of these do we want to address significantly better than our competitors?

How, at a high level, do we plan to solve them?

Will customers say, 'This is special, I'd be delighted to pay more'?

See The Value Curve.

MESSAGING

How will we communicate our Value Proposition to the customers?

What are the benefits that customers would derive from our key features? How can we prove that?

What stories and emotions does our product evoke?

KEY METRICS

Define a few key metrics to measure how your product is doing and whether the strategy is working.

Consider the North Star Metric and One Metric That Matters (OMTM)..

GROWTH

How do we envision growth? Is it PLG or Sales-Led Growth?

What are our preferred Sales and Marketing channels?

Will we rely on Social Media, SEO, Influencers, or Resellers?

ASK YOURSELF

What makes us think competitors can’t or won’t copy our strategy?

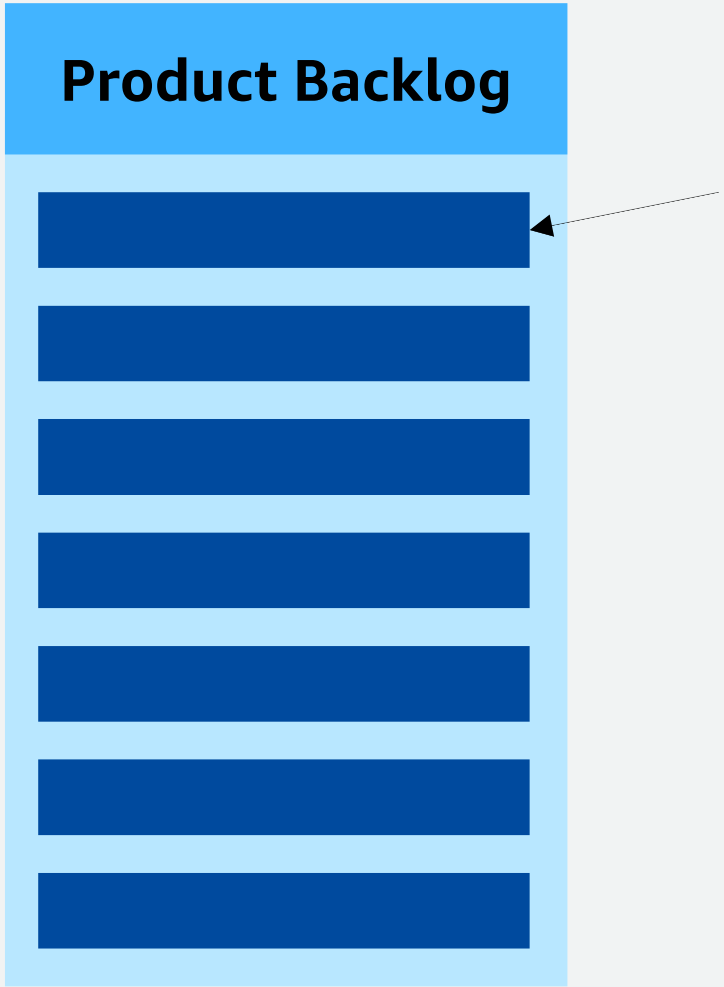
Do the various elements of our strategy fit together and reinforce each other?

What needs to be true for this strategy to work? How can we validate these assumptions?

Source: https://www.productcompass.pm/p/product-strategy-canvas

aws

Product Backlog and User Stories



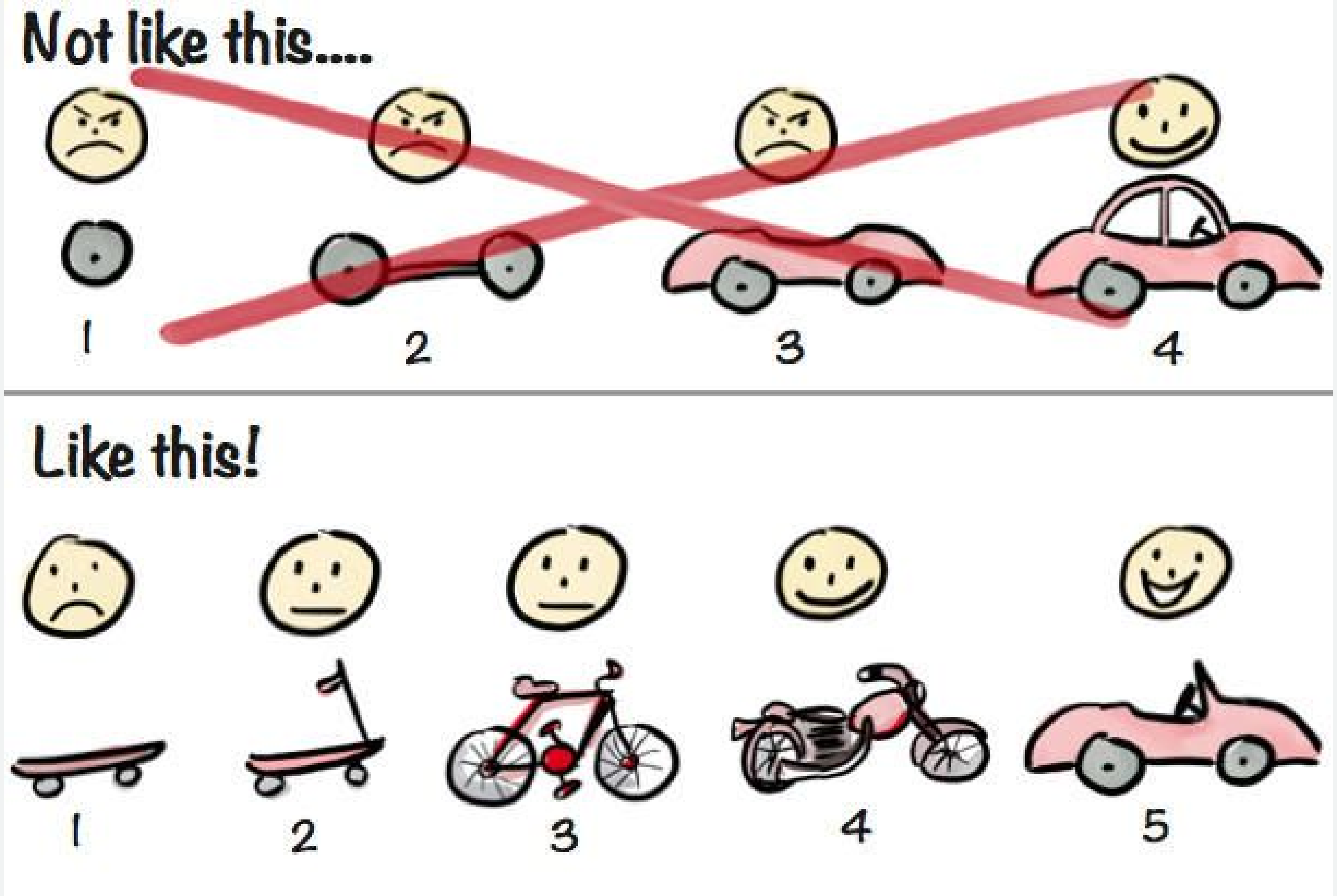
“As a [user role],
I want [goal/desire]
so that [benefit/value].”

As an online shopper, I want to be able to filter products by size and color, so that I can quickly find items that match my preferences.

Acceptance Criteria:

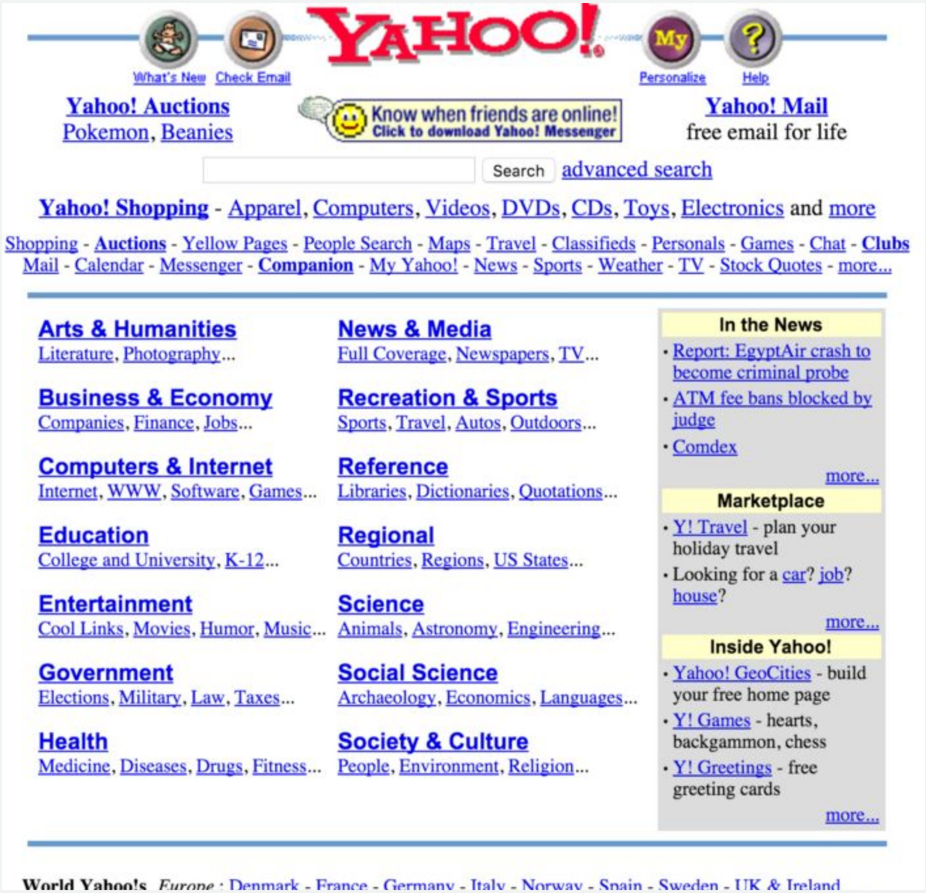
- *The filter options for size and color should be visible on the product listing page.*
- *Filtering by size should show only products available in the selected size(s).*
- *Filtering by color should show only products available in the selected color(s).*
- *It should be possible to combine size and color filtersThe number of results matching the applied filters should be displayed.*

Minimum Viable Product



Source: https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp

Features vs. Value



Yahoo! Auctions
Pokemon, Beanies

Know when friends are online!
Click to download Yahoo! Messenger

Yahoo! Mail
free email for life

advanced search

Yahoo! Shopping - Apparel, Computers, Videos, DVDs, CDs, Toys, Electronics and more

Shopping - **Auctions** - Yellow Pages - People Search - Maps - Travel - Classifieds - Personals - Games - Chat - **Clubs**
Mail - Calendar - Messenger - **Companion** - My Yahoo! - News - Sports - Weather - TV - Stock Quotes - more...

Arts & Humanities
Literature, Photography...

Business & Economy
Companies, Finance, Jobs...

Computers & Internet
Internet, WWW, Software, Games...

Education
College and University, K-12...

Entertainment
Cool Links, Movies, Humor, Music...

Government
Elections, Military, Law, Taxes...

Health
Medicine, Diseases, Drugs, Fitness...

News & Media
Full Coverage, Newspapers, TV...

Recreation & Sports
Sports, Travel, Autos, Outdoors...

Reference
Libraries, Dictionaries, Quotations...

Regional
Countries, Regions, US States...

Science
Animals, Astronomy, Engineering...

Social Science
Archaeology, Economics, Languages...

Society & Culture
People, Environment, Religion...

In the News

- Report: EgyptAir crash to become criminal probe
- ATM fee bans blocked by judge
- Comdex

more...

Marketplace

- Y! Travel - plan your holiday travel
- Looking for a car? job? house?

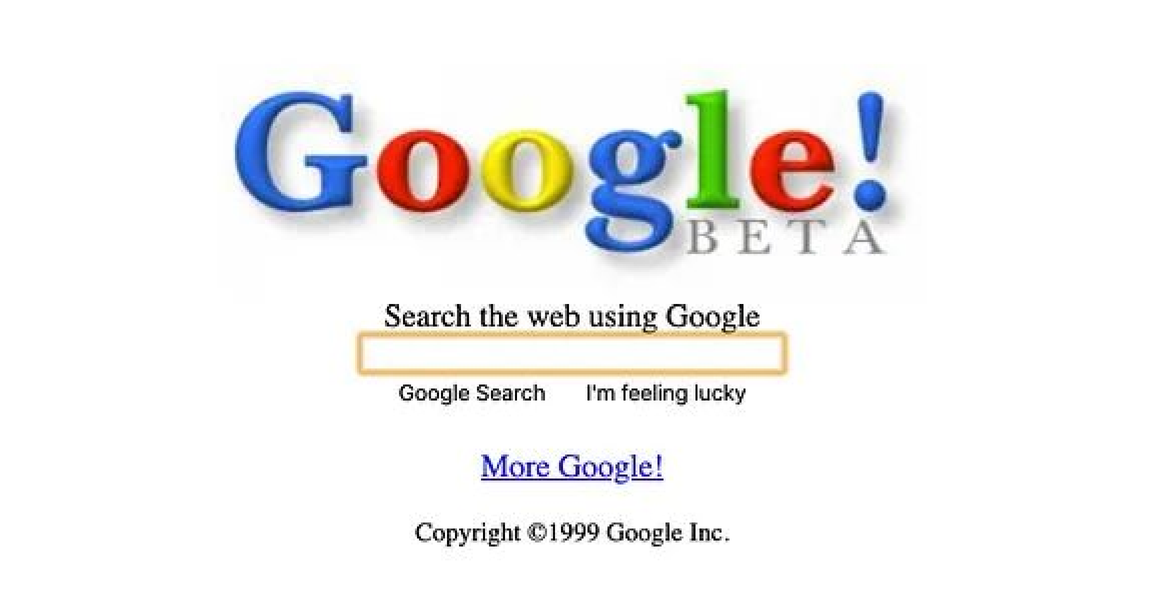
more...

Inside Yahoo!

- Yahoo! GeoCities - build your free home page
- Y! Games - hearts, backgammon, chess
- Y! Greetings - free greeting cards

more...

World Yahoo!s *Europe* : Denmark - France - Germany - Italy - Norway - Spain - Sweden - UK & Ireland

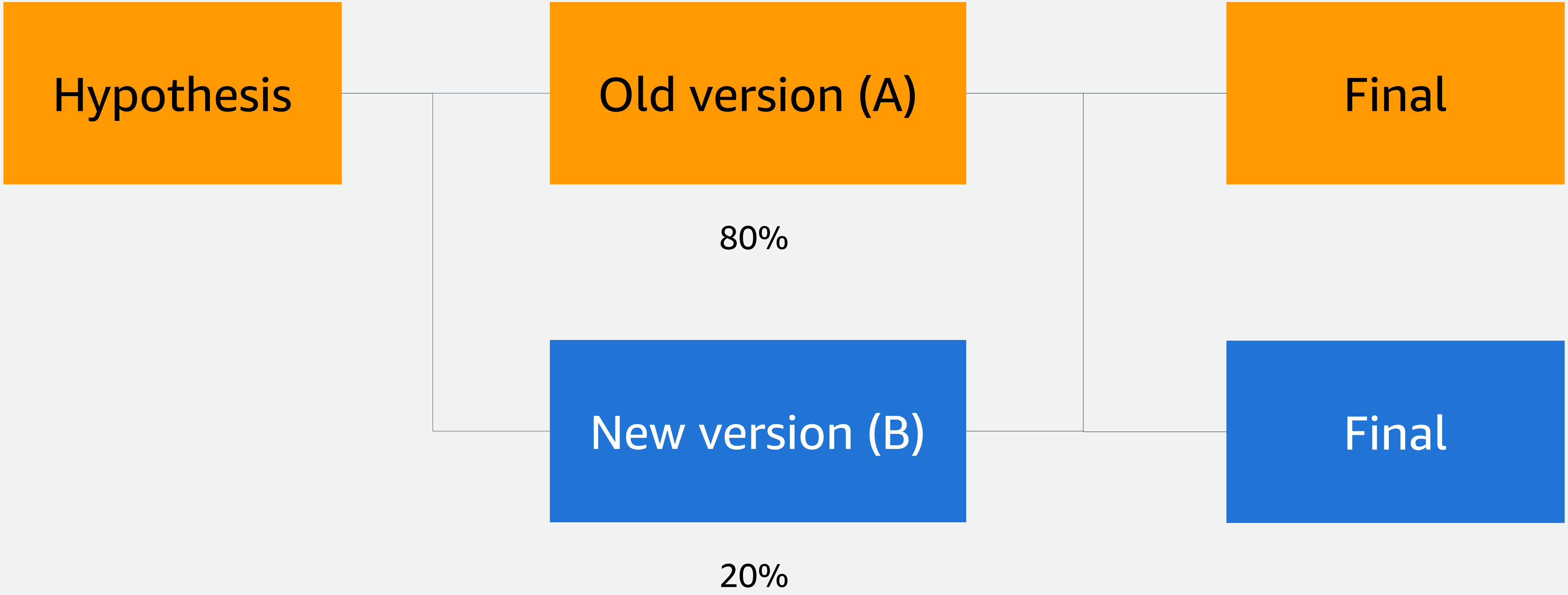


Search the web using Google

More Google!

Copyright ©1999 Google Inc.

Hypothesis-driven Development



Hypothesis

Old version (A)

80%

New version (B)

20%

Final

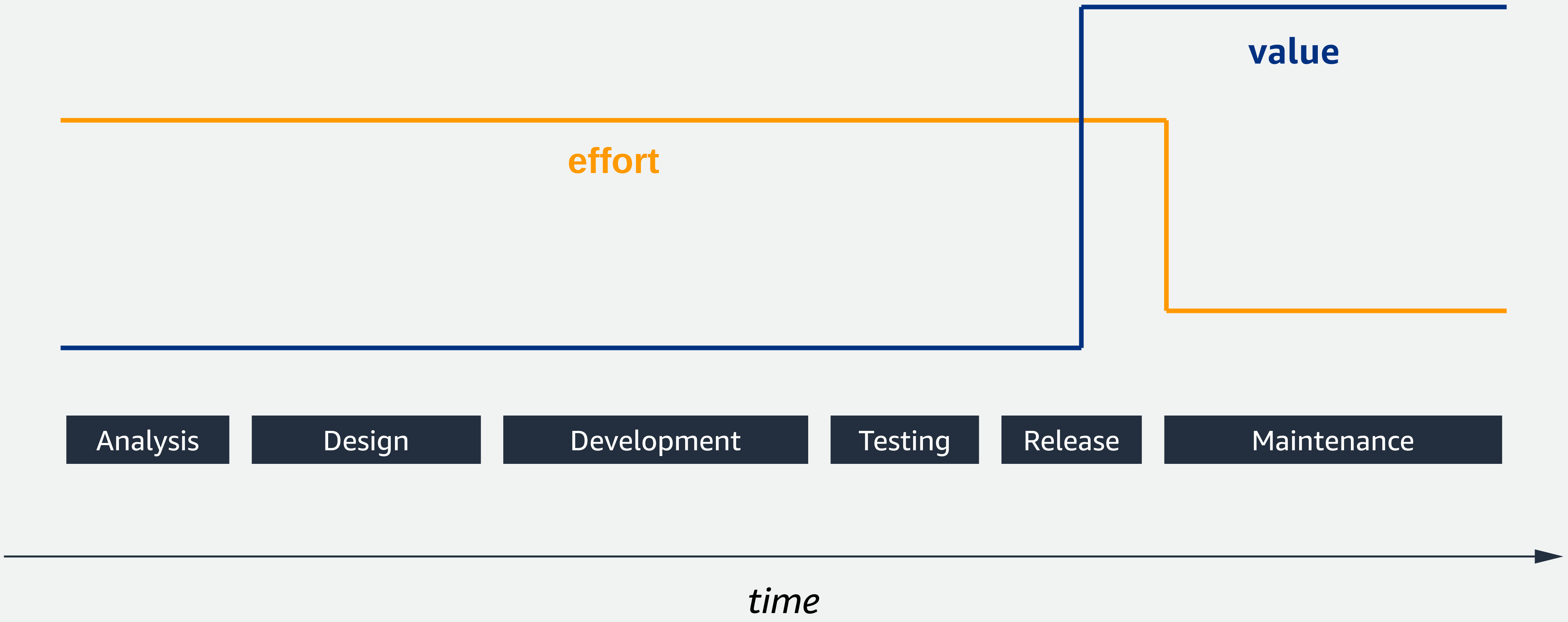
Final

Strategic Metrics

Business	Sales strategy alignment	[✓ Revenue Growth]	Ethical Business Practices	Mergers and Acquisitions	Break-Even Point	Profitability Index (PI)	Reputational Risk	Financial Health	Partnerships and Alliances	Cannibalisation Risk
	Business model alignment	Serviceable Obtainable Market (SOM)	Financial stability	Time to recognize revenue	Cost Performance Index (CPI)	Payback Period	Impact on Core Business	Operational Capacity	Resource Utilisation Rate	Fixed Cost Ratio
	Market leadership	Serviceable Available Market (SAM)	[✓ Market Growth Rate]	Minimize Financial Losses	Legal and Regulatory compliance	[✓ Return on Investment (ROI)]	Risks Management	Sales by Region	Customer segmentation	Variable Cost Ratio
	Market Share	Total Addressable Market (TAM)	Market Expansion	New Client Acquisition rate	Count of non-compliant incidents	Profit Margin	Costs Efficiency	Average Revenue per User (ARPU)	Product / Capability Performance	Brand Equity
Product	GTM strategy alignment	Industry innovator recognition	Profitability	Cross and Up-selling opportunity	Total Cost of Ownership (TCO)	Customer Engagement Score	[✓ Customer Lifetime value]	Qualified Marketing traffic	Growth and scalability	Fraud losses
	Marketing strategy alignment	Brand Awareness	Costs efficiency	[✓ Long-term Viability]	Diversification	Activation rate	Customer Acquisition costs	Expansion revenue	Ecosystem development	Chargeback rate
	Market fit	Differentiation	Customer Satisfaction score	Number of active users	Leads by Lifecycle stage	Costs per feature	Depth of usage	Lead-to-Customer rate	Customer churn	CAC-to-LTV Ratio
	Lifecycle Management	User Engagement	Customer Retention Rate	Seat utilisation rate	Payback period	Customer Health score	Costs per tenant	Costs per transaction	Revenue churn	Net Promoter score

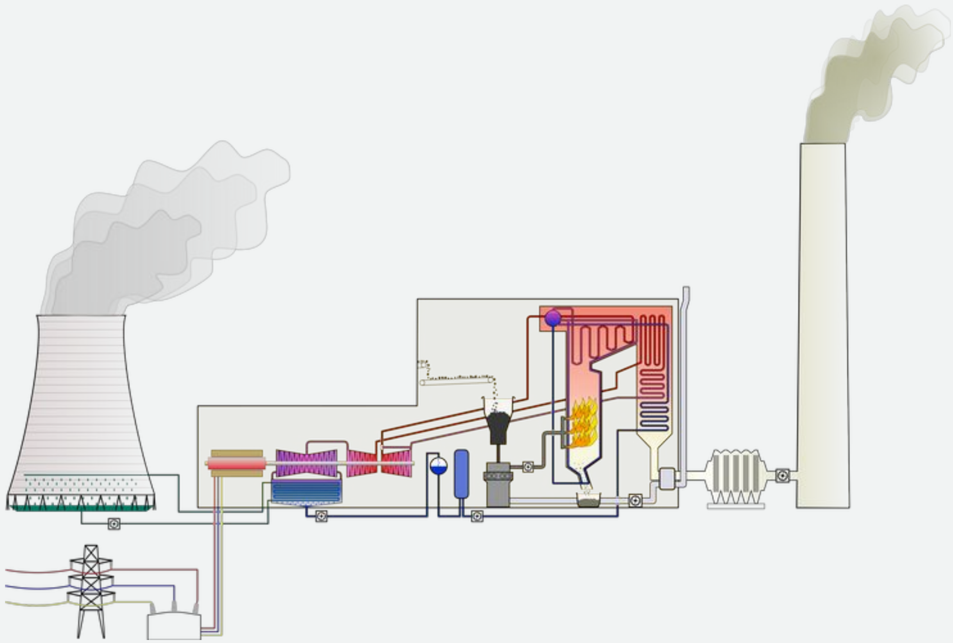
DevOps and Agility

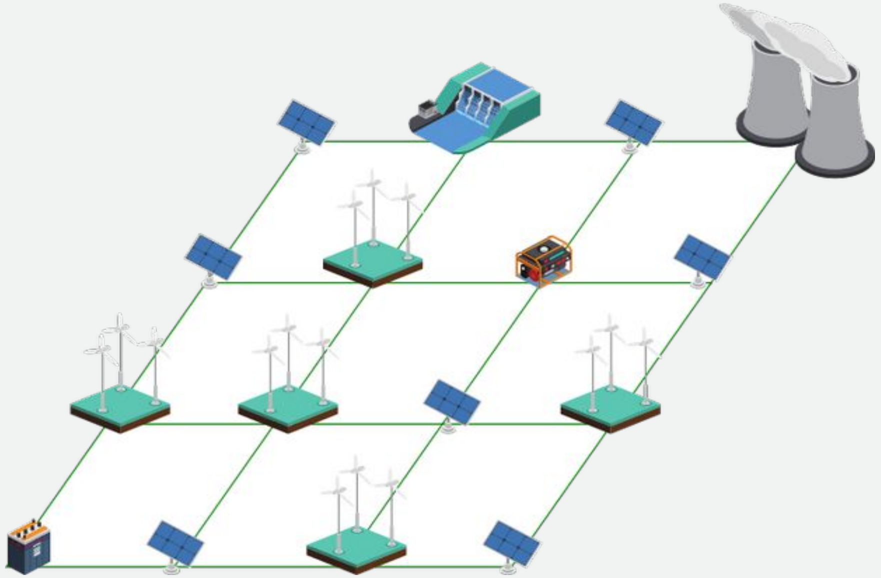
Traditional (“Waterfall”) Lifecycle



“Agile development is adaptive rather than predictive.”

– Martin Fowler





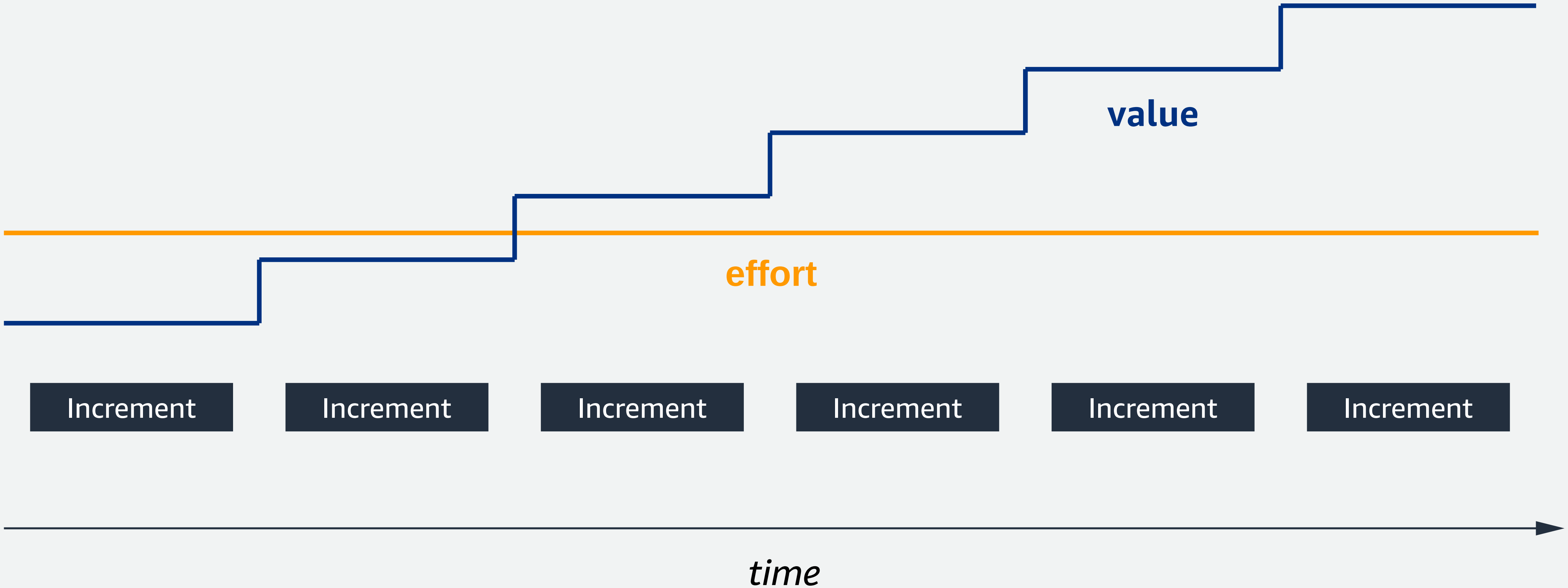
PREDICTIVE APPROACH

- Upfront planning, unknow demand.
- Very difficult to scale.
- Highly complex, hard to maintain.
- Tight coupling, high interdependence.

ADAPTIVE APPROACH

- Start small, expand incrementally.
- Easy to scale and adapt.
- Relatively little complexity.
- Loose coupling, low interdependence.

Agile Lifecycle



The Agile Manifesto

“**Individuals and interactions** over processes and tools.
Working software over comprehensive documentation.
Customer collaboration over contract negotiation.
Responding to change over following a plan.”

– Agile Manifesto [https://agilemanifesto.org/]

The Agile Manifesto – 12 Principles

“Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.”

– Agile Principles (excerpt) [https://agilemanifesto.org/principles.html]

The Agile Manifesto – 12 Principles

“Build projects around motivated individuals.
Give them the environment and support they need,
and trust them to get the job done.”

– Agile Principles (excerpt) [https://agilemanifesto.org/principles.html]

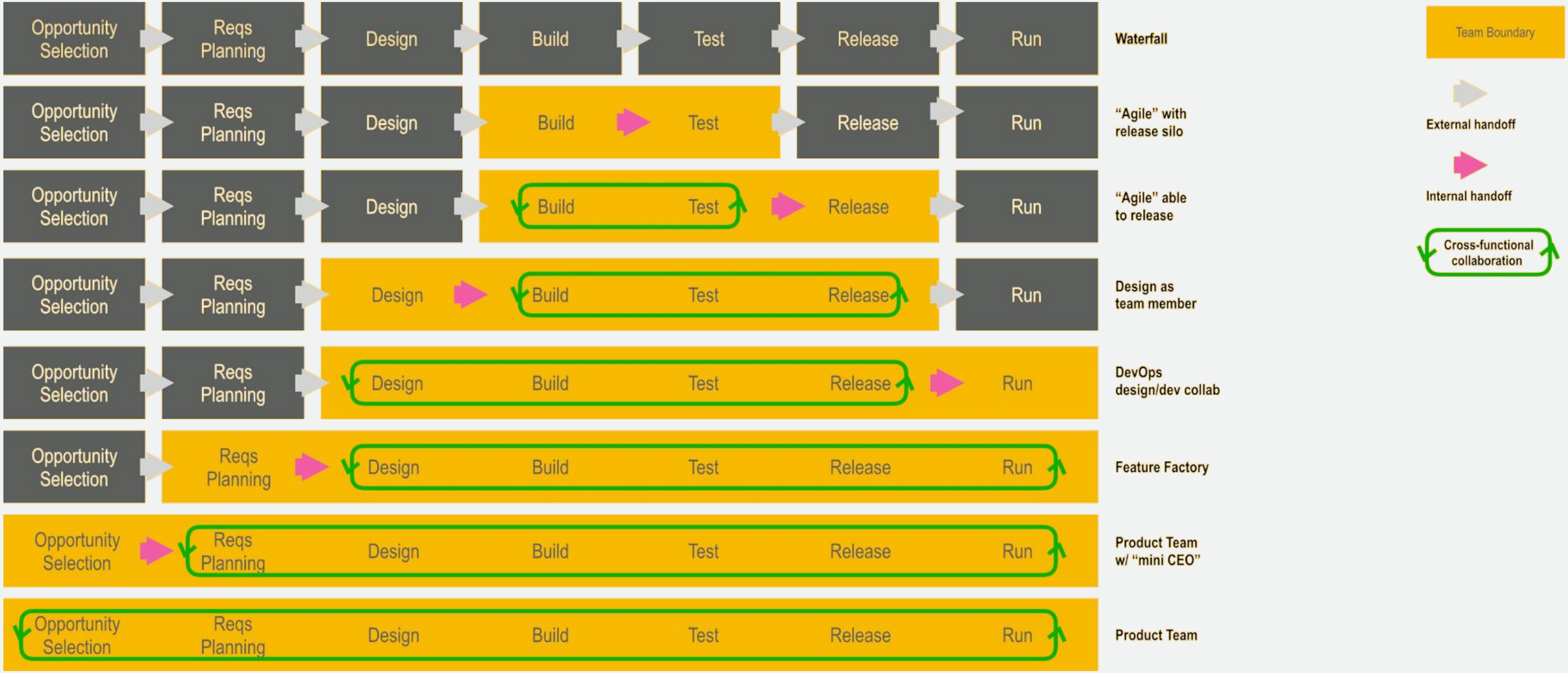
The Agile Manifesto – 12 Principles

“Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”

– Agile Principles (excerpt) [https://agilemanifesto.org/principles.html]

DevOps Organisation

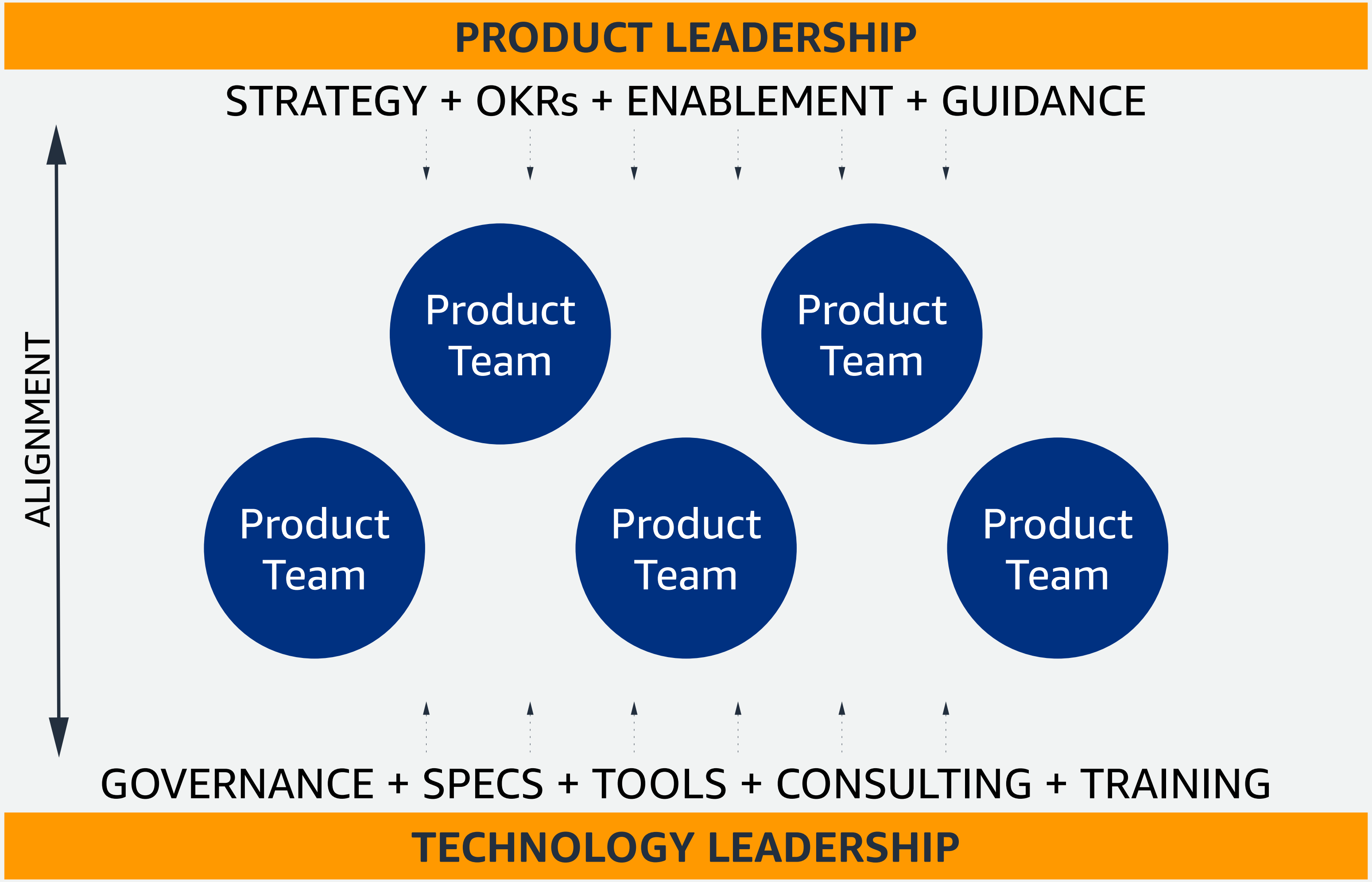
End-to-end Responsibility



Opportunity Selection	Reqs Planning	Design	Build	Test	Release	Run	Waterfall
Opportunity Selection	Reqs Planning	Design	Build	Test	Release	Run	"Agile" with release silo
Opportunity Selection	Reqs Planning	Design	Build	Test	Release	Run	"Agile" able to release
Opportunity Selection	Reqs Planning	Design	Build	Test	Release	Run	Design as team member
Opportunity Selection	Reqs Planning	Design	Build	Test	Release	Run	DevOps design/dev collab
Opportunity Selection	Reqs Planning	Design	Build	Test	Release	Run	Feature Factory
Opportunity Selection	Reqs Planning	Design	Build	Test	Release	Run	Product Team w/ "mini CEO"
Opportunity Selection	Reqs Planning	Design	Build	Test	Release	Run	Product Team

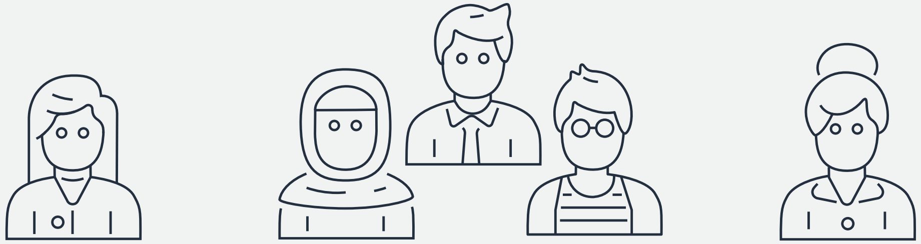
Source: https://amplitude.com/blog/journey-to-product-teams-infographic

LATERAL LEADERSHIP



Scrum (2002)

Team

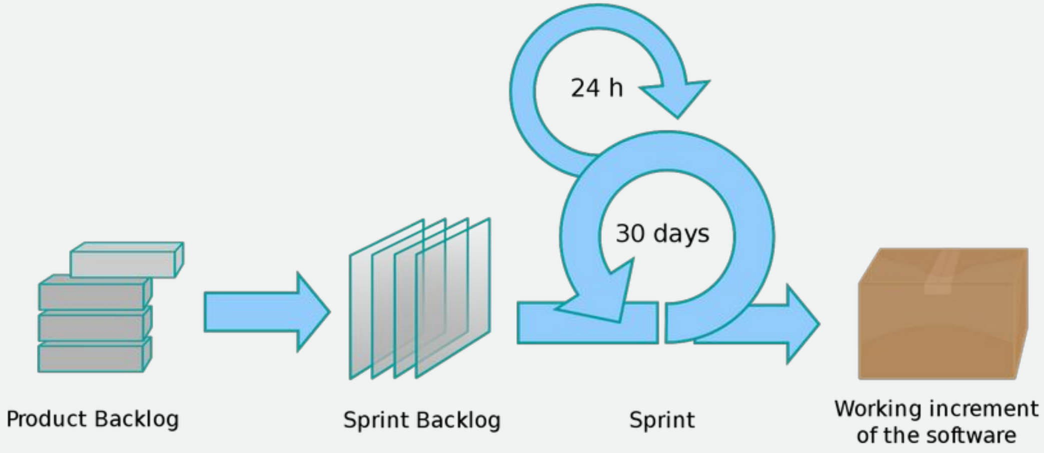


Product Owner

Development Team

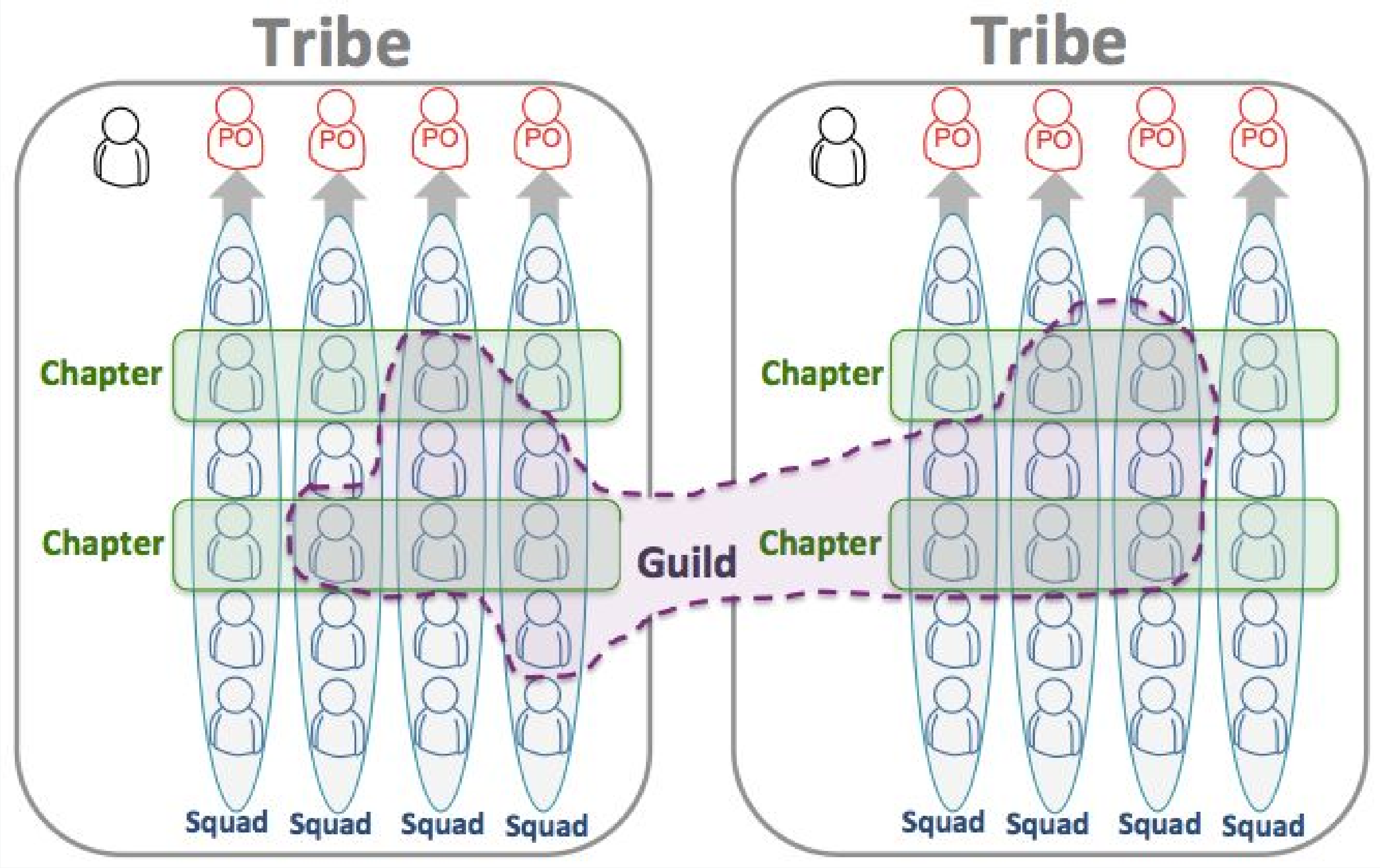
Scrum Master

Process



Source: https://en.wikipedia.org/wiki/Scrum_(software_development)

Spotify Model (2012)



Source: *https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf*

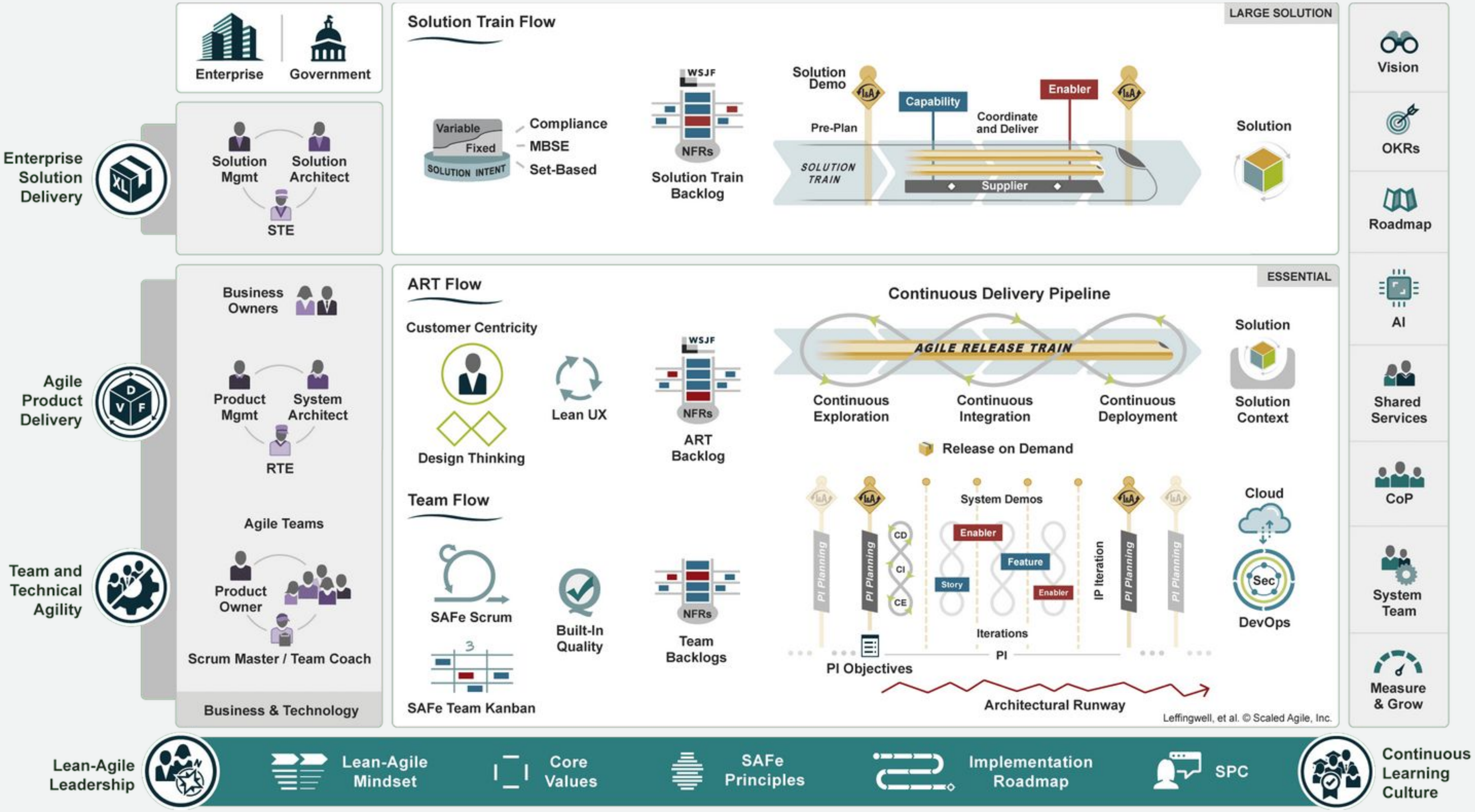
Governance?

Consistency?

Accountability?

Culture?

SAFe Framework (2015)



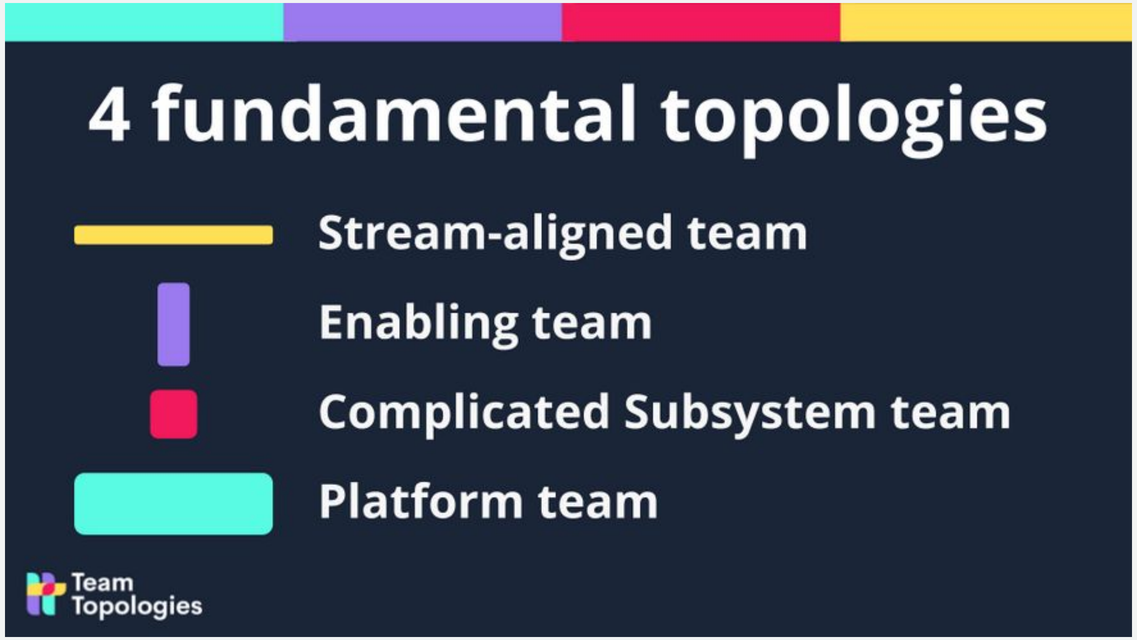
"People over processes"?

"Responding to change"?

"Build projects around motivated individuals"?

Source: https://scaledagileframework.com

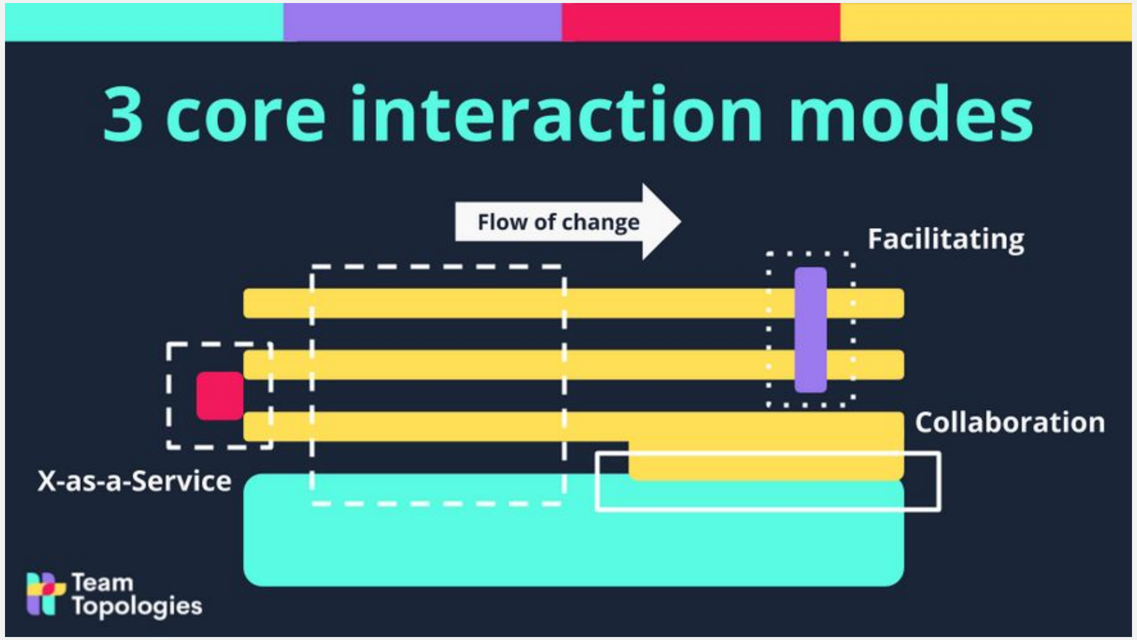
Team Topologies (2019)



4 fundamental topologies

- 🟨 Stream-aligned team
- 🟪 Enabling team
- 🟥 Complicated Subsystem team
- 🟩 Platform team

Team Topologies

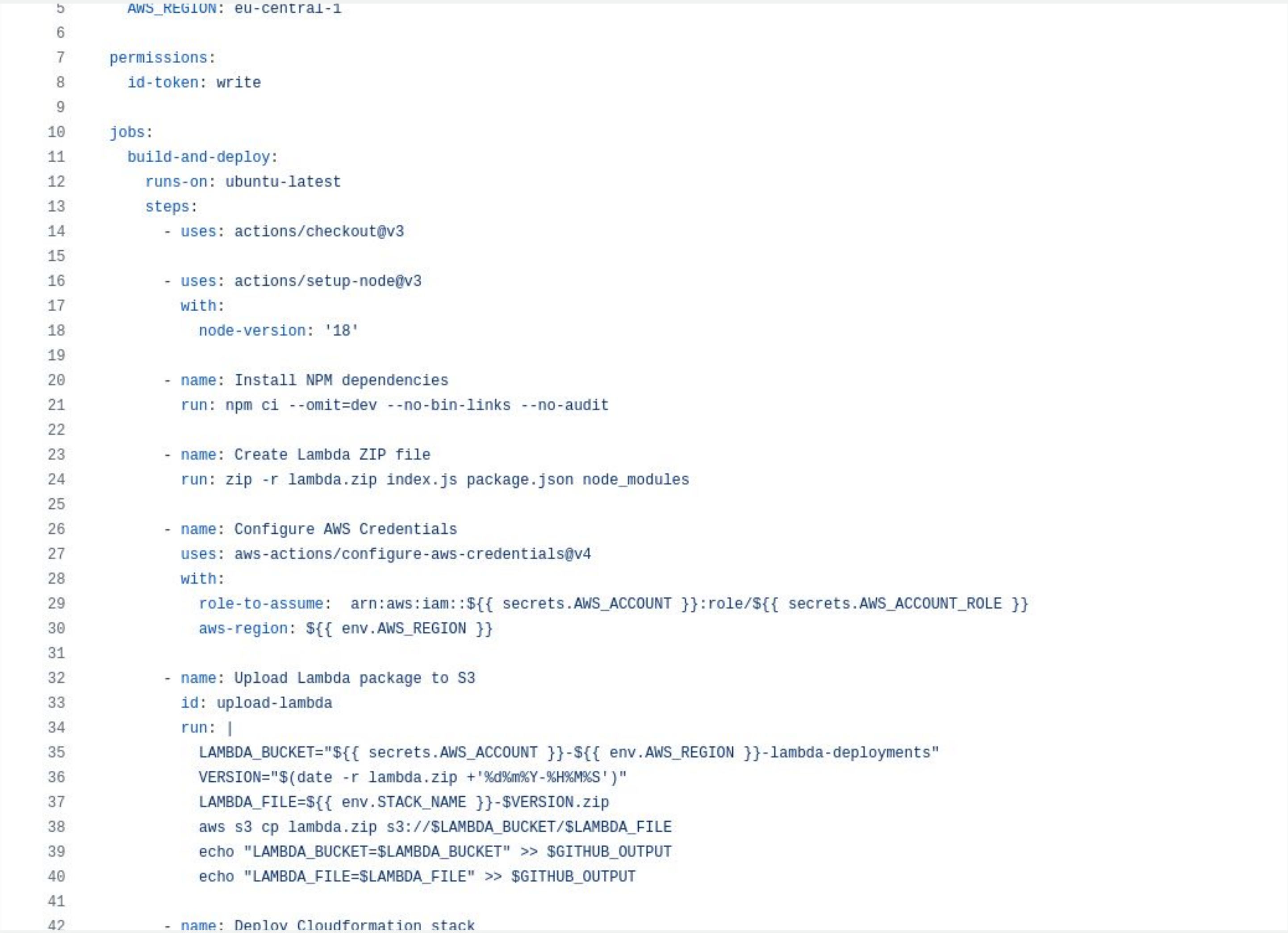


3 core interaction modes

Source: *https://teamtopologies.com/key-concepts*

DevOps Tools and Practices

Continuous Integration and Continuous Delivery (CI/CD)



```yaml
  AWS_REGION: eu-central-1

permissions:
  id-token: write

jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3

      - uses: actions/setup-node@v3
        with:
          node-version: '18'

      - name: Install NPM dependencies
        run: npm ci --omit=dev --no-bin-links --no-audit

      - name: Create Lambda ZIP file
        run: zip -r lambda.zip index.js package.json node_modules

      - name: Configure AWS Credentials
        uses: aws-actions/configure-aws-credentials@v4
        with:
          role-to-assume:  arn:aws:iam::${{ secrets.AWS_ACCOUNT }}:role/${{ secrets.AWS_ACCOUNT_ROLE }}
          aws-region: ${{ env.AWS_REGION }}

      - name: Upload Lambda package to S3
        id: upload-lambda
        run: |
          LAMBDA_BUCKET="${{ secrets.AWS_ACCOUNT }}-${{ env.AWS_REGION }}-lambda-deployments"
          VERSION="$(date -r lambda.zip +'%d%m%Y-%H%M%S')"
          LAMBDA_FILE=${{ env.STACK_NAME }}-$VERSION.zip
          aws s3 cp lambda.zip s3://$LAMBDA_BUCKET/$LAMBDA_FILE
          echo "LAMBDA_BUCKET=$LAMBDA_BUCKET" >> $GITHUB_OUTPUT
          echo "LAMBDA_FILE=$LAMBDA_FILE" >> $GITHUB_OUTPUT

      - name: Deploy Cloudformation stack
```

Checkout repo

(Code linting)

Build

(Unit tests)

(Integration tests)

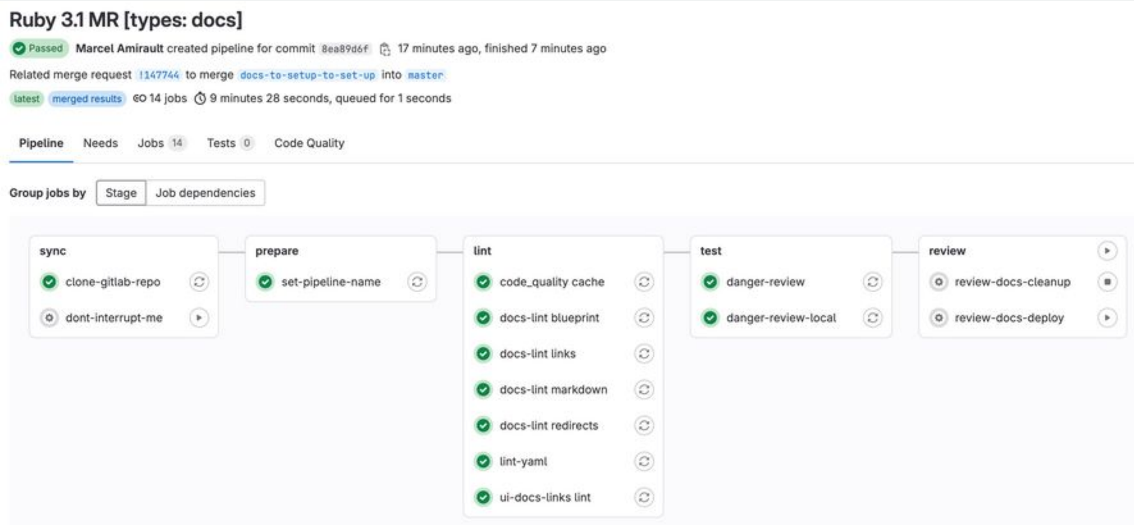
(Functional tests)

(Security tests)

(Packaging)

Delivery/Deployment

Continuous Integration and Continuous Delivery (CI/CD)



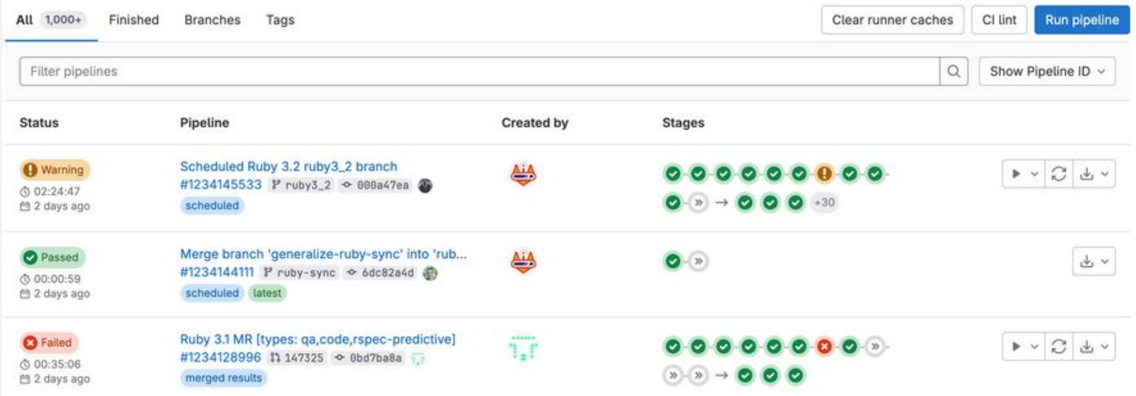
Ruby 3.1 MR [types: docs]

Passed **Marcel Amirault** created pipeline for commit `8ea89d6f` 17 minutes ago, finished 7 minutes ago

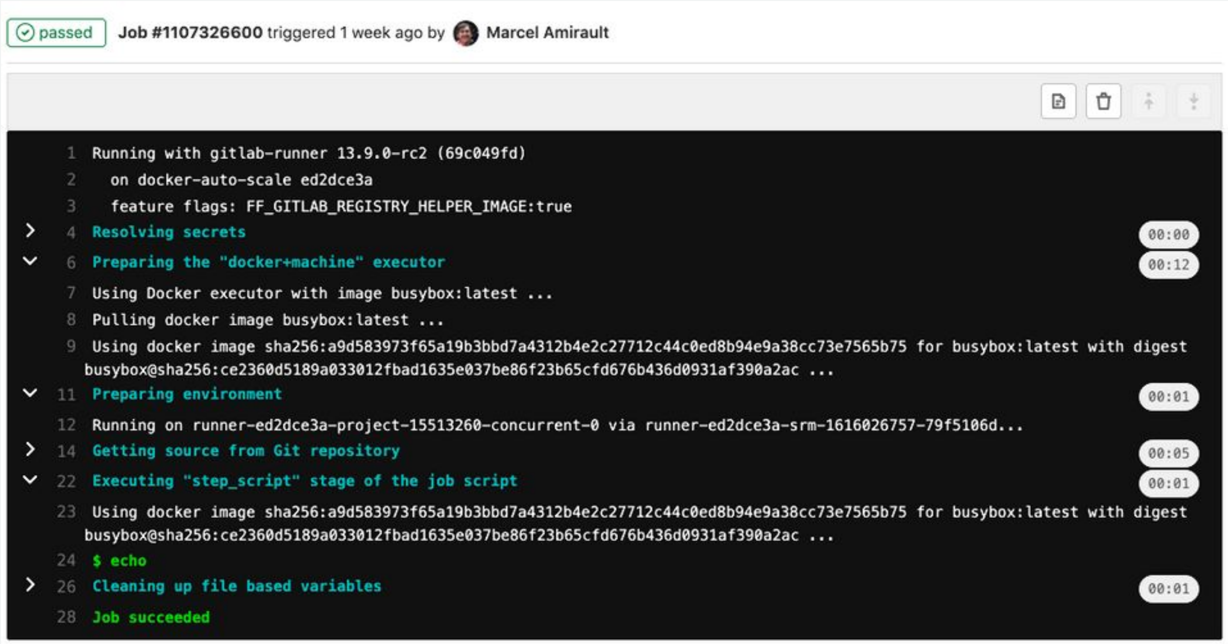
Related merge request `!147744` to merge `docs-to-setup-to-set-up` into `master`

latest merged results 14 jobs 9 minutes 28 seconds, queued for 1 seconds

sync	prepare	lint	test	review
clone-gitlab-repo	set-pipeline-name	code_quality cache	danger-review	review-docs-cleanup
dont-interrupt-me		docs-lint blueprint	danger-review-local	review-docs-deploy
		docs-lint links		
		docs-lint markdown		
		docs-lint redirects		
		lint-yaml		
		ui-docs-links lint		



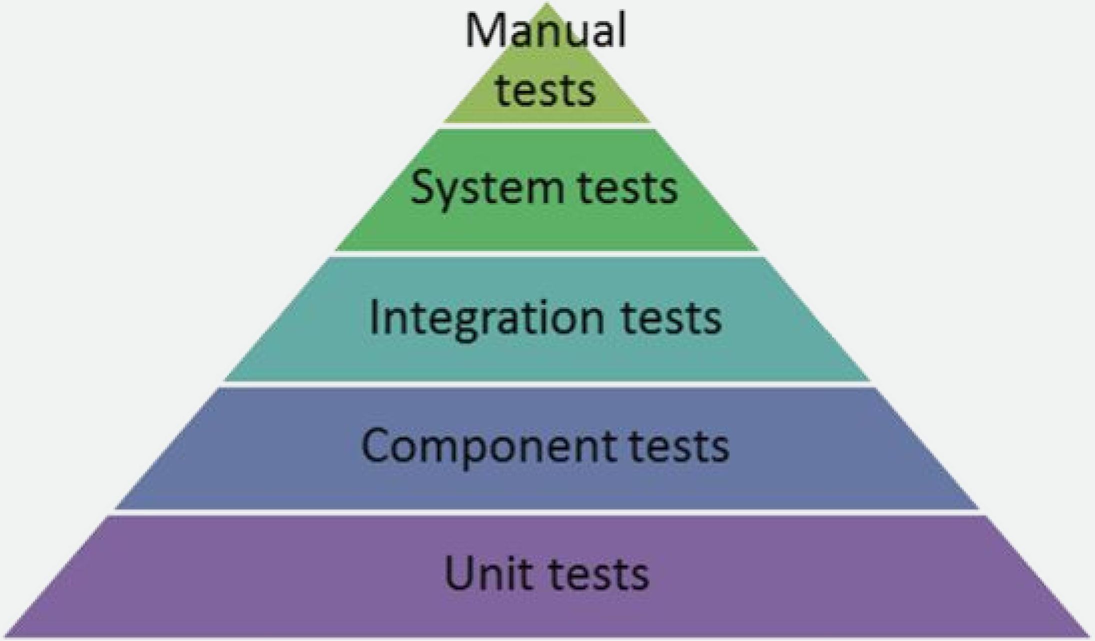
Status	Pipeline	Created by	Stages
Warning 02:24:47 2 days ago	Scheduled Ruby 3.2 ruby3_2 branch #1234145533 ruby3_2 000a47ea scheduled		✓ ✓ ✓ ✓ ✓ ✓ ! ✓ ✓ ✓ » → ✓ ✓ ✓ +30
Passed 00:00:59 2 days ago	Merge branch 'generalize-ruby-sync' into 'rub... #1234144111 ruby-sync 6dc82a4d scheduled latest		✓ »
Failed 00:35:06 2 days ago	Ruby 3.1 MR [types: qa,code,rspec-predictive] #1234128996 147325 0bd7ba8a merged results		✓ ✓ ✓ ✓ ✓ ✓ ✗ ✓ » » » → ✓ ✓ ✓

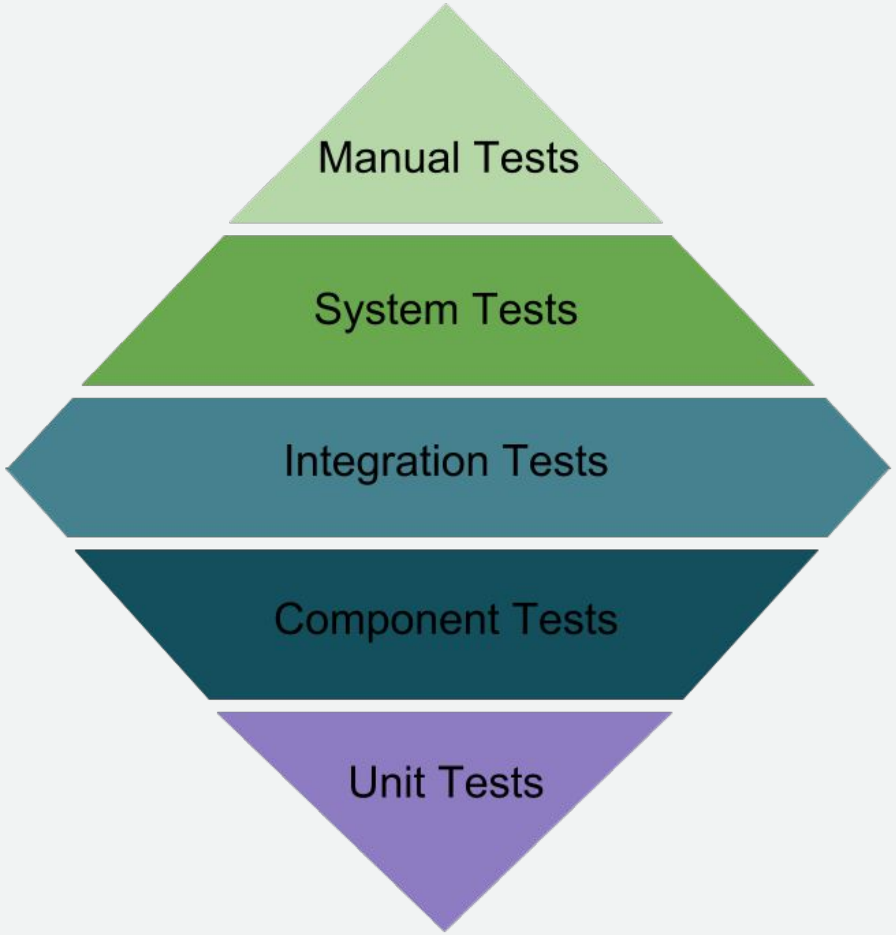


passed **Job #1107326600** triggered 1 week ago by **Marcel Amirault**

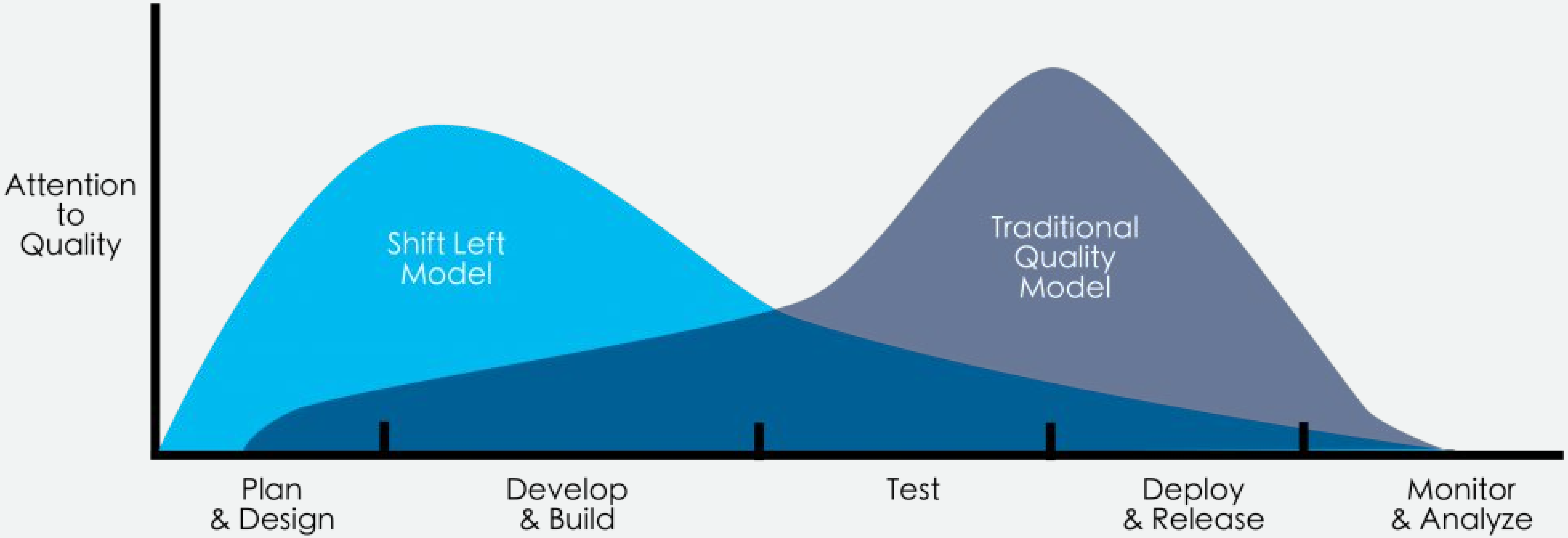
```
Running with gitlab-runner 13.9.0-rc2 (69c049fd)
  on docker-auto-scale ed2dce3a
  feature flags: FF_GITLAB_REGISTRY_HELPER_IMAGE:true
Resolving secrets                                                     00:00
Preparing the "docker+machine" executor                               00:12
Using Docker executor with image busybox:latest ...
Pulling docker image busybox:latest ...
Using docker image sha256:a9d583973f65a19b3bbd7a4312b4e2c27712c44c0ed8b94e9a38cc73e7565b75 for busybox:latest with digest
busybox@sha256:ce2360d5189a033012fbad1635e037be86f23b65cfd676b436d0931af390a2ac ...
Preparing environment                                                 00:01
Running on runner-ed2dce3a-project-15513260-concurrent-0 via runner-ed2dce3a-srm-1616026757-79f5106d...
Getting source from Git repository                                    00:05
Executing "step_script" stage of the job script                       00:01
Using docker image sha256:a9d583973f65a19b3bbd7a4312b4e2c27712c44c0ed8b94e9a38cc73e7565b75 for busybox:latest with digest
busybox@sha256:ce2360d5189a033012fbad1635e037be86f23b65cfd676b436d0931af390a2ac ...
$ echo
Cleaning up file based variables                                      00:01
Job succeeded
```

Testing Strategies



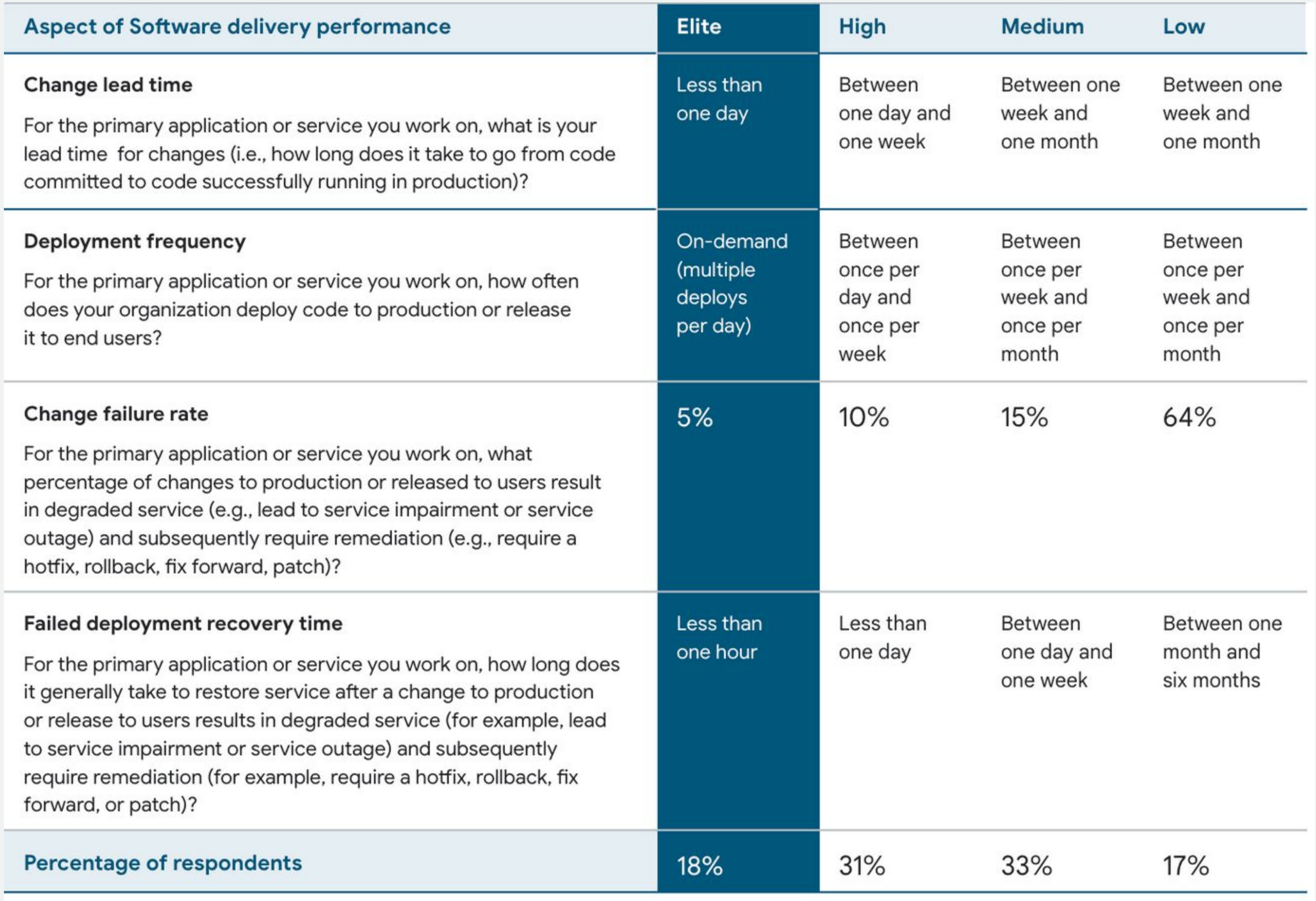


Shift Left



Source: https://devopedia.org/shift-left

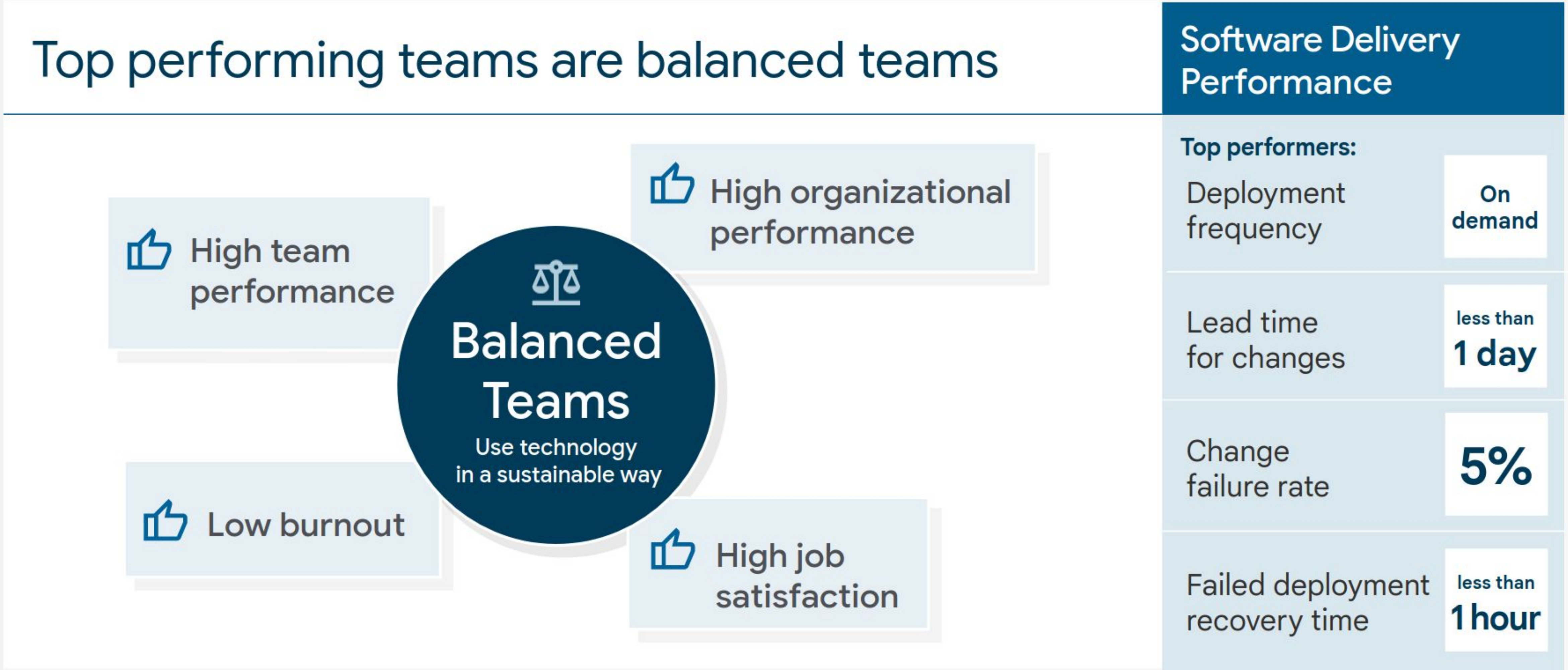
DORA Metrics



Aspect of Software delivery performance	Elite	High	Medium	Low
Change lead time For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one week and one month
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per week and once per month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	5%	10%	15%	64%
Failed deployment recovery time For the primary application or service you work on, how long does it generally take to restore service after a change to production or release to users results in degraded service (for example, lead to service impairment or service outage) and subsequently require remediation (for example, require a hotfix, rollback, fix forward, or patch)?	Less than one hour	Less than one day	Between one day and one week	Between one month and six months
Percentage of respondents	18%	31%	33%	17%

Source: https://cloud.google.com/blog/products/ devops-sre/announcing-the-2023-state-of-devops-

High-performing Teams



Top performing teams are balanced teams

High team performance

High organizational performance

Balanced Teams

Use technology in a sustainable way

Low burnout

High job satisfaction

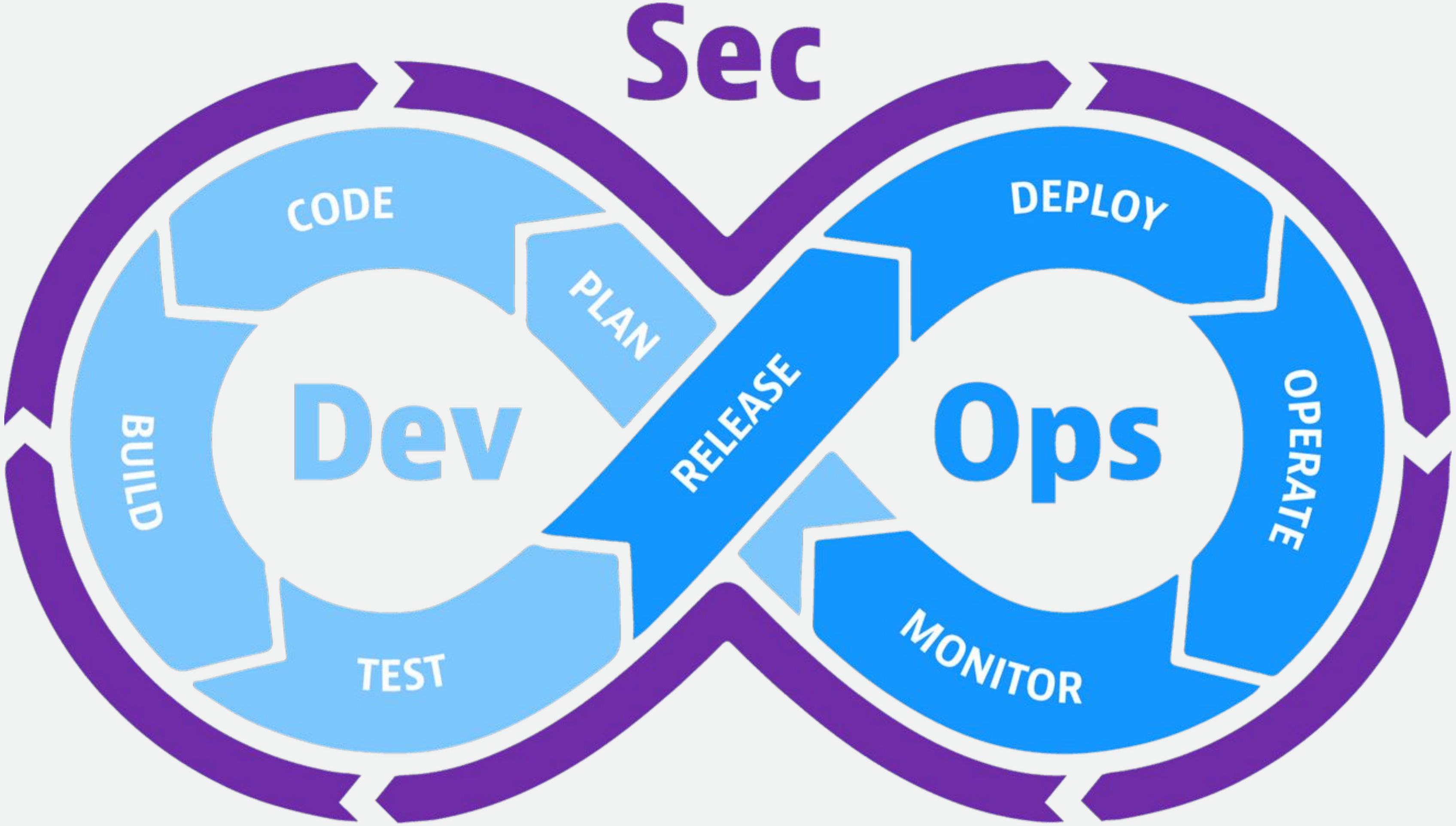
Software Delivery Performance

Top performers:

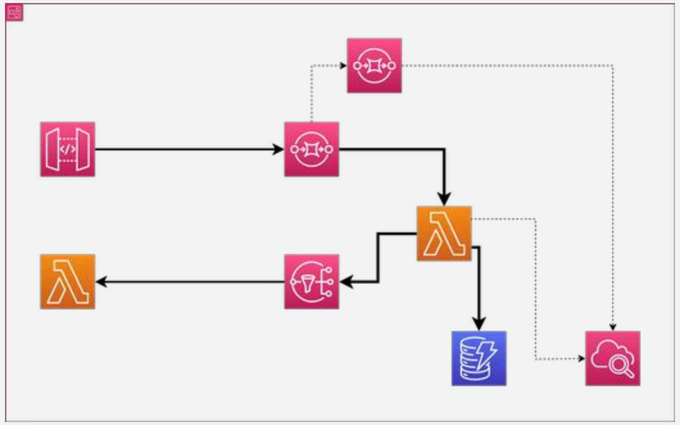
Deployment frequency	**On demand**
Lead time for changes	less than **1 day**
Change failure rate	**5%**
Failed deployment recovery time	less than **1 hour**

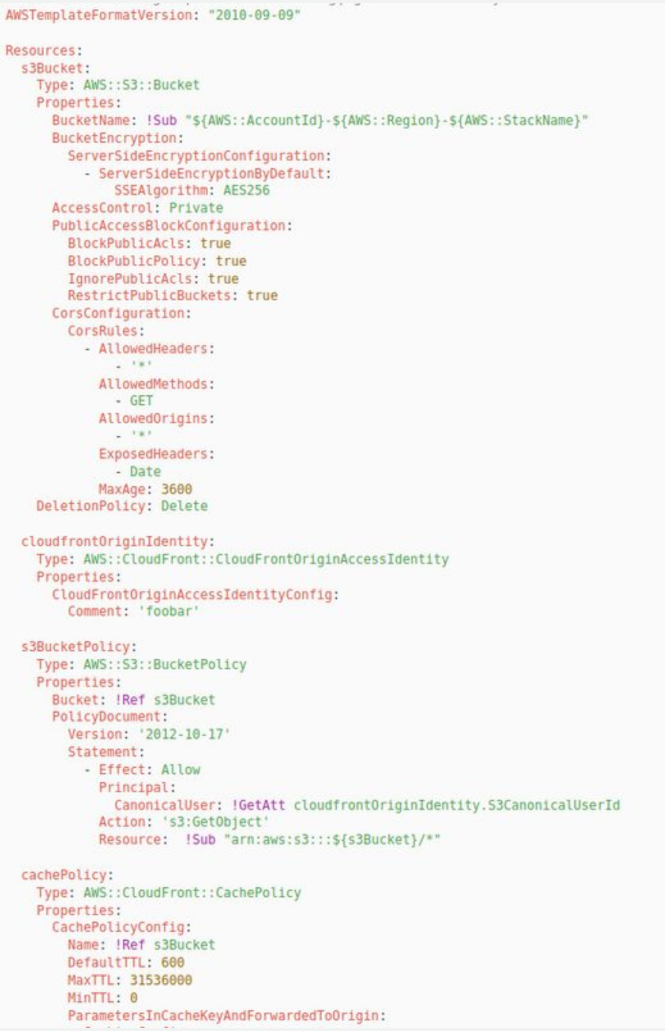
Source: https://cloud.google.com/blog/products/devops-sre/announcing-the-2023-state-of-devops-report

DevSecOps

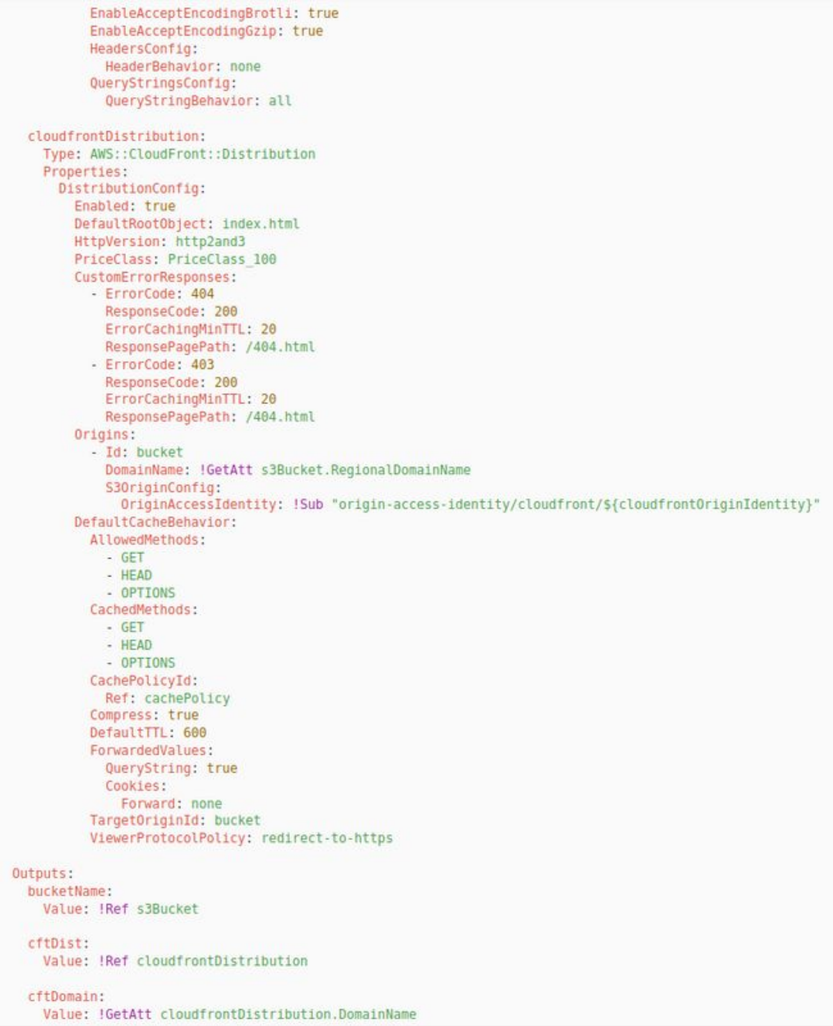

Sec
CODE
PLAN
BUILD
Dev
TEST
RELEASE
DEPLOY
Ops
OPERATE
MONITOR

Infrastructure as Code



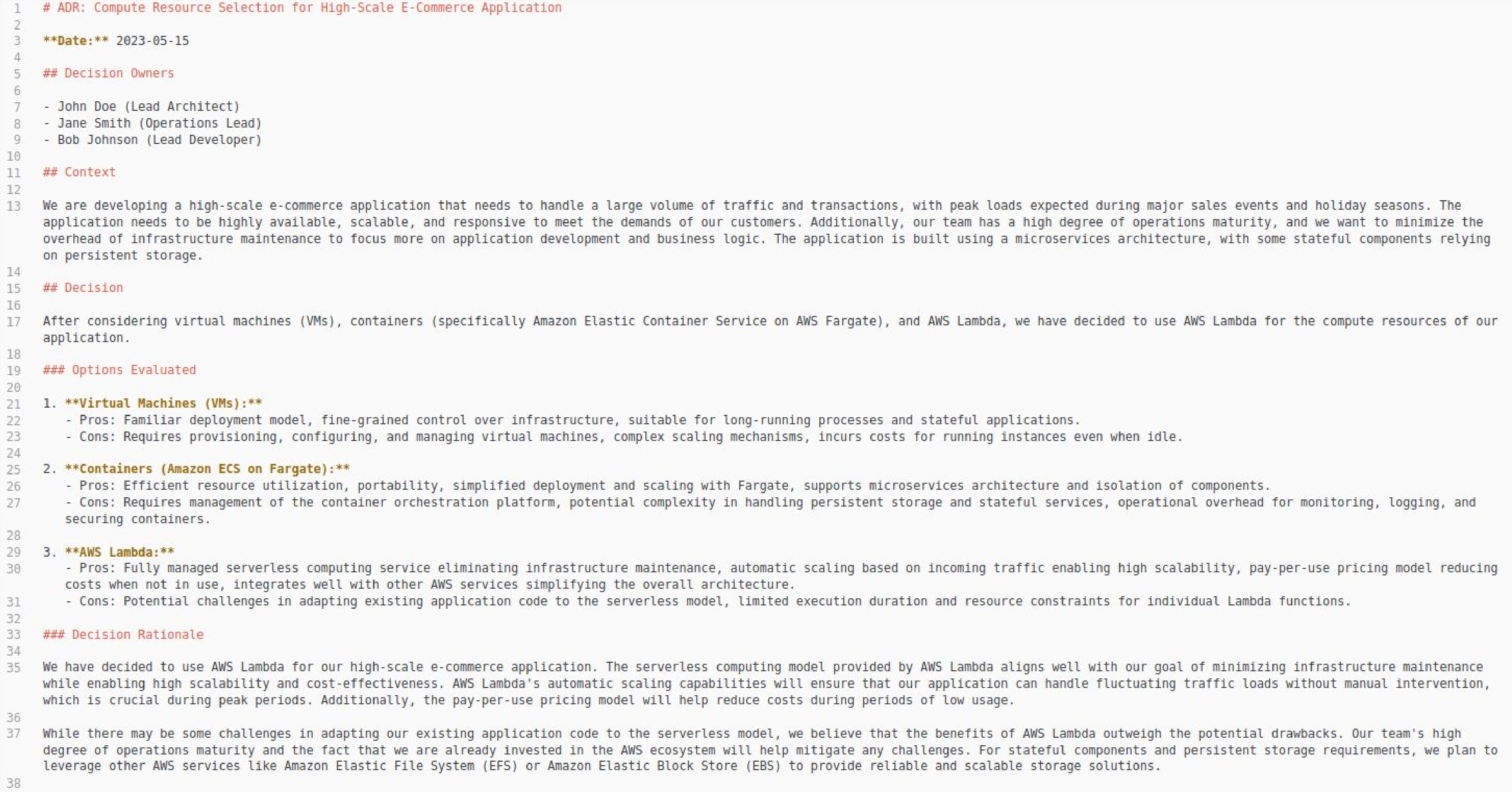


```yaml
AWSTemplateFormatVersion: "2010-09-09"

Resources:
  s3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Sub "${AWS::AccountId}-${AWS::Region}-${AWS::StackName}"
      BucketEncryption:
        ServerSideEncryptionConfiguration:
          - ServerSideEncryptionByDefault:
              SSEAlgorithm: AES256
      AccessControl: Private
      PublicAccessBlockConfiguration:
        BlockPublicAcls: true
        BlockPublicPolicy: true
        IgnorePublicAcls: true
        RestrictPublicBuckets: true
      CorsConfiguration:
        CorsRules:
          - AllowedHeaders:
              - '*'
            AllowedMethods:
              - GET
            AllowedOrigins:
              - '*'
            ExposedHeaders:
              - Date
            MaxAge: 3600
    DeletionPolicy: Delete

  cloudfrontOriginIdentity:
    Type: AWS::CloudFront::CloudFrontOriginAccessIdentity
    Properties:
      CloudFrontOriginAccessIdentityConfig:
        Comment: 'foobar'

  s3BucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket: !Ref s3Bucket
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              CanonicalUser: !GetAtt cloudfrontOriginIdentity.S3CanonicalUserId
            Action: 's3:GetObject'
            Resource:  !Sub "arn:aws:s3:::${s3Bucket}/*"

  cachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        Name: !Ref s3Bucket
        DefaultTTL: 600
        MaxTTL: 31536000
        MinTTL: 0
        ParametersInCacheKeyAndForwardedToOrigin:
```



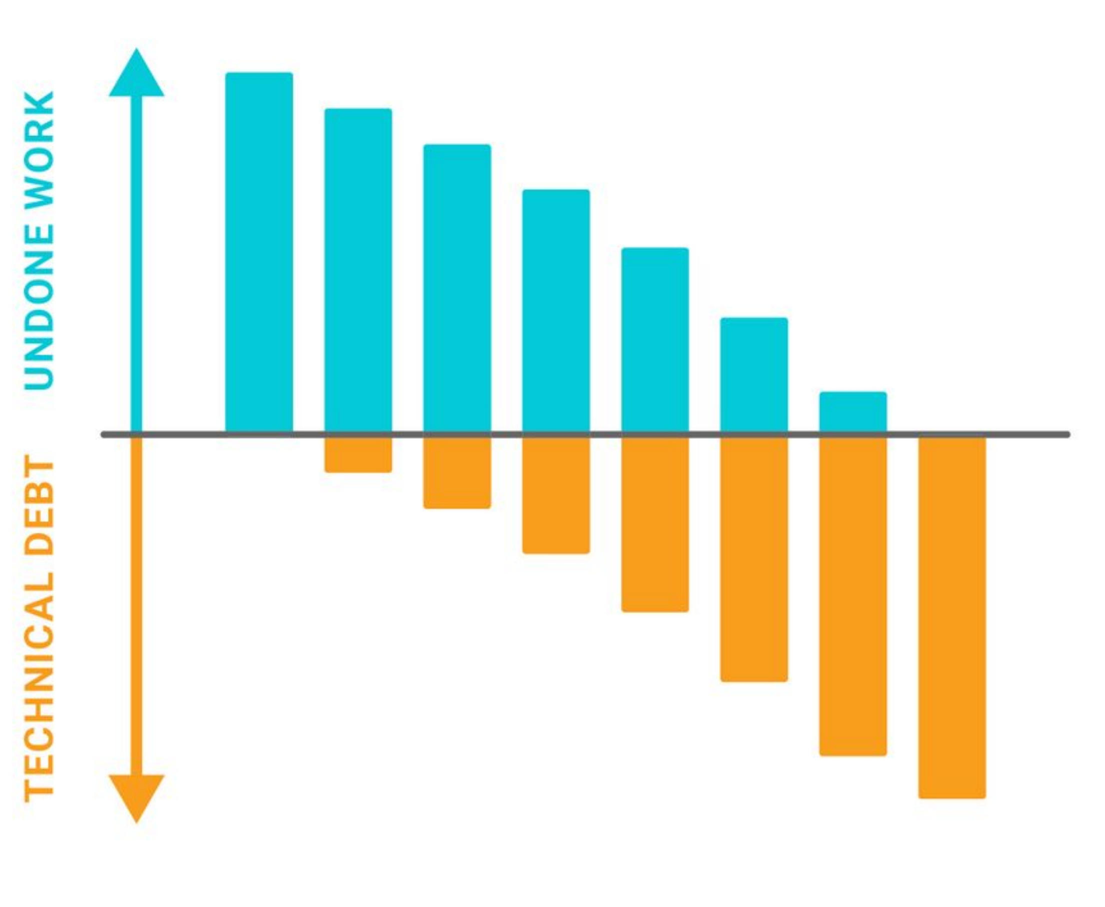
```yaml
          EnableAcceptEncodingBrotli: true
          EnableAcceptEncodingGzip: true
          HeadersConfig:
            HeaderBehavior: none
          QueryStringsConfig:
            QueryStringBehavior: all

  cloudfrontDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Enabled: true
        DefaultRootObject: index.html
        HttpVersion: http2and3
        PriceClass: PriceClass_100
        CustomErrorResponses:
          - ErrorCode: 404
            ResponseCode: 200
            ErrorCachingMinTTL: 20
            ResponsePagePath: /404.html
          - ErrorCode: 403
            ResponseCode: 200
            ErrorCachingMinTTL: 20
            ResponsePagePath: /404.html
        Origins:
          - Id: bucket
            DomainName: !GetAtt s3Bucket.RegionalDomainName
            S3OriginConfig:
              OriginAccessIdentity: !Sub "origin-access-identity/cloudfront/${cloudfrontOriginIdentity}"
        DefaultCacheBehavior:
          AllowedMethods:
            - GET
            - HEAD
            - OPTIONS
          CachedMethods:
            - GET
            - HEAD
            - OPTIONS
          CachePolicyId:
            Ref: cachePolicy
          Compress: true
          DefaultTTL: 600
          ForwardedValues:
            QueryString: true
            Cookies:
              Forward: none
          TargetOriginId: bucket
          ViewerProtocolPolicy: redirect-to-https

Outputs:
  bucketName:
    Value: !Ref s3Bucket

  cftDist:
    Value: !Ref cloudfrontDistribution

  cftDomain:
    Value: !GetAtt cloudfrontDistribution.DomainName
```

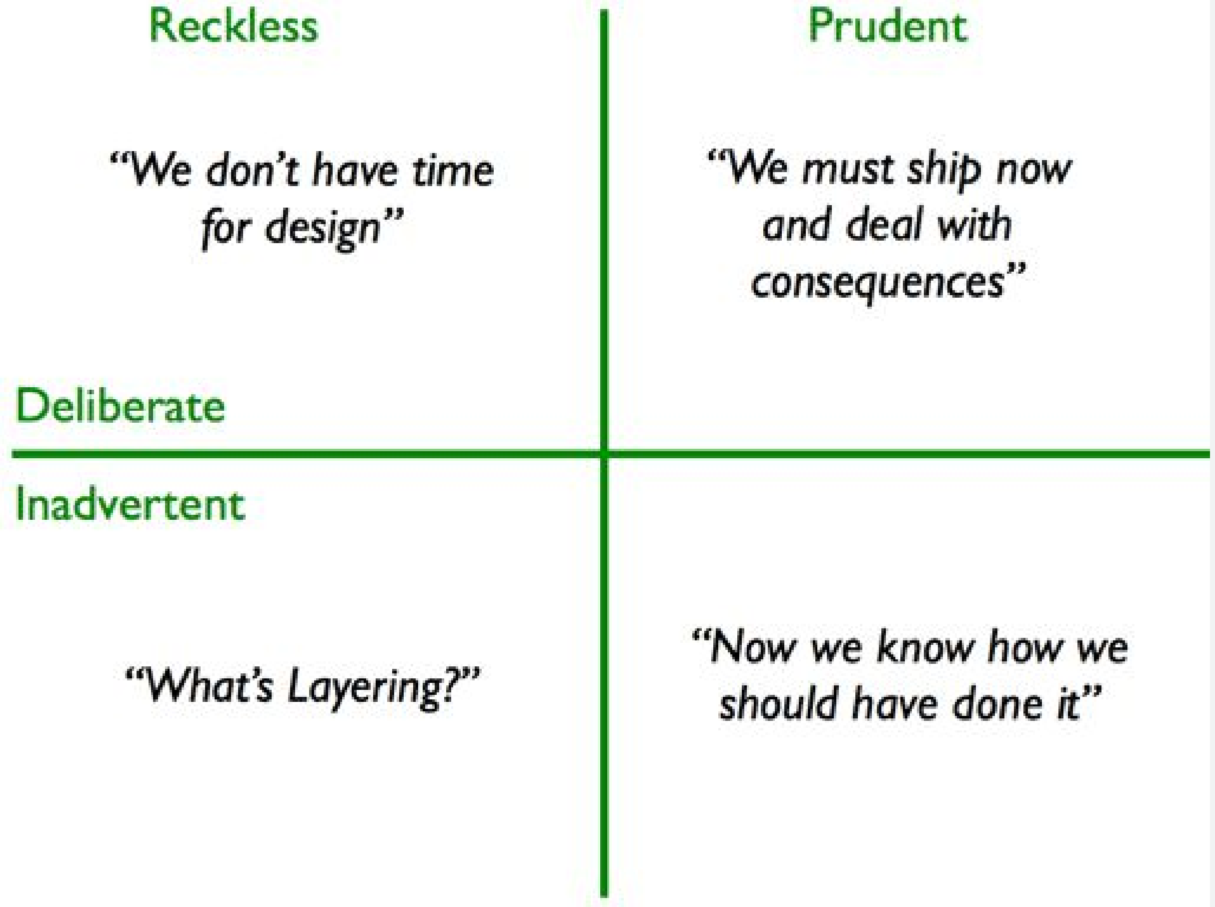
Architecture Decision Records



```markdown
# ADR: Compute Resource Selection for High-Scale E-Commerce Application

**Date:** 2023-05-15

## Decision Owners

- John Doe (Lead Architect)
- Jane Smith (Operations Lead)
- Bob Johnson (Lead Developer)

## Context

We are developing a high-scale e-commerce application that needs to handle a large volume of traffic and transactions, with peak loads expected during major sales events and holiday seasons. The application needs to be highly available, scalable, and responsive to meet the demands of our customers. Additionally, our team has a high degree of operations maturity, and we want to minimize the overhead of infrastructure maintenance to focus more on application development and business logic. The application is built using a microservices architecture, with some stateful components relying on persistent storage.

## Decision

After considering virtual machines (VMs), containers (specifically Amazon Elastic Container Service on AWS Fargate), and AWS Lambda, we have decided to use AWS Lambda for the compute resources of our application.

### Options Evaluated

1. **Virtual Machines (VMs):**
   - Pros: Familiar deployment model, fine-grained control over infrastructure, suitable for long-running processes and stateful applications.
   - Cons: Requires provisioning, configuring, and managing virtual machines, complex scaling mechanisms, incurs costs for running instances even when idle.

2. **Containers (Amazon ECS on Fargate):**
   - Pros: Efficient resource utilization, portability, simplified deployment and scaling with Fargate, supports microservices architecture and isolation of components.
   - Cons: Requires management of the container orchestration platform, potential complexity in handling persistent storage and stateful services, operational overhead for monitoring, logging, and securing containers.

3. **AWS Lambda:**
   - Pros: Fully managed serverless computing service eliminating infrastructure maintenance, automatic scaling based on incoming traffic enabling high scalability, pay-per-use pricing model reducing costs when not in use, integrates well with other AWS services simplifying the overall architecture.
   - Cons: Potential challenges in adapting existing application code to the serverless model, limited execution duration and resource constraints for individual Lambda functions.

### Decision Rationale

We have decided to use AWS Lambda for our high-scale e-commerce application. The serverless computing model provided by AWS Lambda aligns well with our goal of minimizing infrastructure maintenance while enabling high scalability and cost-effectiveness. AWS Lambda's automatic scaling capabilities will ensure that our application can handle fluctuating traffic loads without manual intervention, which is crucial during peak periods. Additionally, the pay-per-use pricing model will help reduce costs during periods of low usage.

While there may be some challenges in adapting our existing application code to the serverless model, we believe that the benefits of AWS Lambda outweigh the potential drawbacks. Our team's high degree of operations maturity and the fact that we are already invested in the AWS ecosystem will help mitigate any challenges. For stateful components and persistent storage requirements, we plan to leverage other AWS services like Amazon Elastic File System (EFS) or Amazon Elastic Block Store (EBS) to provide reliable and scalable storage solutions.
```

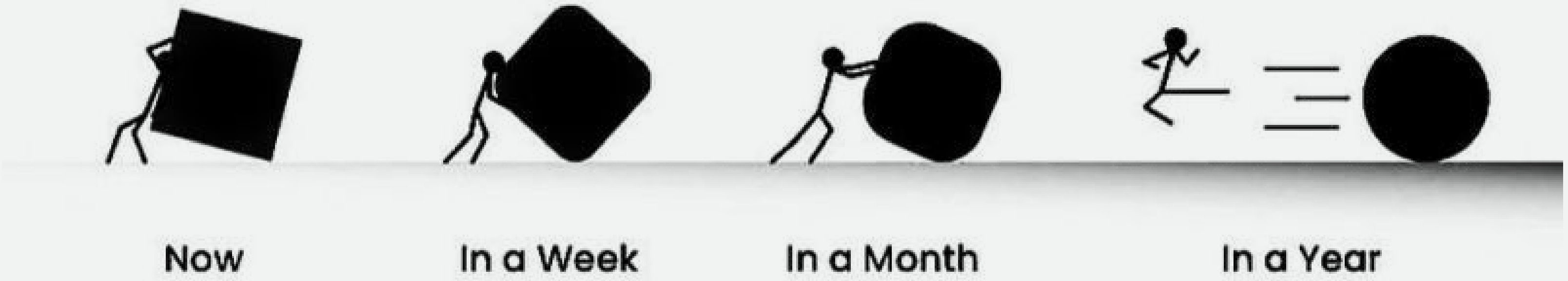
Technical Debt





	Reckless	Prudent
Deliberate	*"We don't have time for design"*	*"We must ship now and deal with consequences"*
Inadvertent	*"What's Layering?"*	*"Now we know how we should have done it"*

Practice forms habits.
Habits form behaviours.



Thank you!

Alexander Günsche

Senior Solutions Architect

lxg@amazon.com