

Ak Way Defaults Cheatsheet v0.4.0



Purpose

Return creativity and joy to the world of software engineering & delivery.

Vision

To be the world's most trusted partner for empowering businesses to accelerate the delivery of their greatest ideas

Mission

Businesses, at any scale, are awash with great ideas. However, the cost of learning which idea customers will love is too high. We evaporate this constraint and build quality in by accelerating cloud native adoption using next-generation consultancy, engineering and immersive learning

Values

- ❑ Honesty
- ❑ Pride
- ❑ Ownership
- ❑ Creativity
- ❑ Diversity

Principles

- ❑ People First
- ❑ Empathy
- ❑ Radical Candor
- ❑ Action Biased
- ❑ Experimentation
- ❑ Fast Feedback

How to use this sheet

This is not a checklist. These are your fallback defaults, if nothing else is set, not a golden master you must do.

Vision & Empathy

We understand the customer

The customer is at the centre

- ❑ Understand the customer
- ❑ Understand our stakeholders
- ❑ Shared view of how value flows to our customers
- ❑ Regular, early and direct feedback from the customer
- ❑ Understand the current domain, language & terminology

We have a clear vision

Highly aligned and focused

- ❑ Understand the broader business's vision
- ❑ We can all present a clear vision, mission and purpose
- ❑ We can measure what good looks like
- ❑ We have just enough of a shareable roadmap

Team-First

Team can be successful

We can have sustainable success

- ❑ Delivery expectation doesn't exceed the team's cognitive capacity
- ❑ Roles, and role rotation are discussed and understood
- ❑ Working agreements negotiated with the whole team
- ❑ We are sensing / listening for evolutions of our team types and interactions
- ❑ We have a Team API or clearly defined service offering

Cloud Native

We start with production first

Production-ready from the outset

- ❑ Production deployed to on day one
- ❑ Deployed "Hello World" on Production

We continuously deploy

Pipelines!

- ❑ All deployments are pipelined
- ❑ Build fixed within 10 min
- ❑ All tests run and are reliable
- ❑ Coding standards are checked
- ❑ Delete code that isn't valuable

We automate everything

Everything as Code

- ❑ Environments recreated from only code
- ❑ Environments regularly destroyed and recreated
- ❑ Tests complete quickly

We do day 2 operations on day 1

Day2Ops right from the start

- ❑ Debugging information is fully accessible to engineers
- ❑ Events are fully traceable to requests
- ❑ Observability & other *abilities are a core consideration
- ❑ Service status is monitored
- ❑ Service Level Indicators & Objectives negotiated
- ❑ We can recover from disasters
- ❑ We can upgrade continuously

Code As Craft

Coding practice

Code well, code fast

- ❑ Commit to main / trunk and run integration at least once a day
- ❑ Leave code better than you found it as part of story delivery

We test everything

Automated tests

- ❑ All new functionality has automated tests
- ❑ Bug fixes have automated tests
- ❑ Non-functional requirements are verified with automated tests
- ❑ Test Driven Development is the default practice
- ❑ Tests give confidence to deploy
- ❑ Failing tests fail the pipeline
- ❑ Meets the Rugged Manifesto
- ❑ Tests complete in minutes

Prototype Working Agreements:

Definition of Ready (DoR)

- ❑ The story discussed in an IPM
- ❑ All acceptance criteria defined
- ❑ The story is prioritised
- ❑ Team has estimated the story
- ❑ Broader value defined

Definition of Done (DoD)

- ❑ All tasks on the story are marked as completed
- ❑ All acceptance criteria achieved
- ❑ All commits are tied to the story
- ❑ Story acceptance instructions defined
- ❑ Any infrastructure is IaC

XP & Agility

We set goals every quarter

Inception meetings

- ❑ Goals, non-, anti-goals defined
- ❑ Actors identified
- ❑ Risks & mitigations identified

We work in weekly iterations

Iterations

- ❑ Started with IPM, restated goals
- ❑ Daily standups, terse and quick
- ❑ Stories completed in priority order
- ❑ Work in progress has limit of 1
- ❑ Pair program daily
- ❑ Ended with a demo & retrospective

We plan work as user stories

User stories

- ❑ Customer value is defined
- ❑ Stories sized to 0.5 days work
- ❑ Acceptance criteria are defined
- ❑ Prioritised with the Product Manager
- ❑ Accepted with the Product Manager
- ❑ Conform to Working Agreements
- ❑ Document just enough to proceed

We measure everything

Learn by experimentation

- ❑ A SMART experiment is defined
- ❑ Experiment is run to the agreed scale, duration and criteria
- ❑ Retrospect on if the experiment succeeded & learning generated

Engagement

We enhance team enthusiasm

Growing engineer confidence

- ❑ Retro outcomes are actions
- ❑ Actions are carried out
- ❑ Rotate roles
- ❑ Everyone demos and talks
- ❑ See one, do one, teach one
- ❑ Individual's learning & development is considered

We radiate our success

Visual working

- ❑ Display WIP Board/backlog
- ❑ Display deployment pipeline
- ❑ Display product metrics
- ❑ Show & tell product and WoW

We demonstrate success

Show it's working

- ❑ Leaders & the team can make simple investment decisions
- ❑ Regular, investment to value-focused reporting

We deliver

Safe & Secure delivery

- ❑ Ways of working are always secure and legal
- ❑ Golden source artefacts live on the owner's systems