

DevOps Habits

- Listen to your users
- Measure key performance indicators (KPI)
- Progressively enable and disable features
- Experiment to maximize learning and influence value

HYPOTHESIS

We believe {customer/**business** segment}
wants {product/feature/**service**}
because {**value** prop}

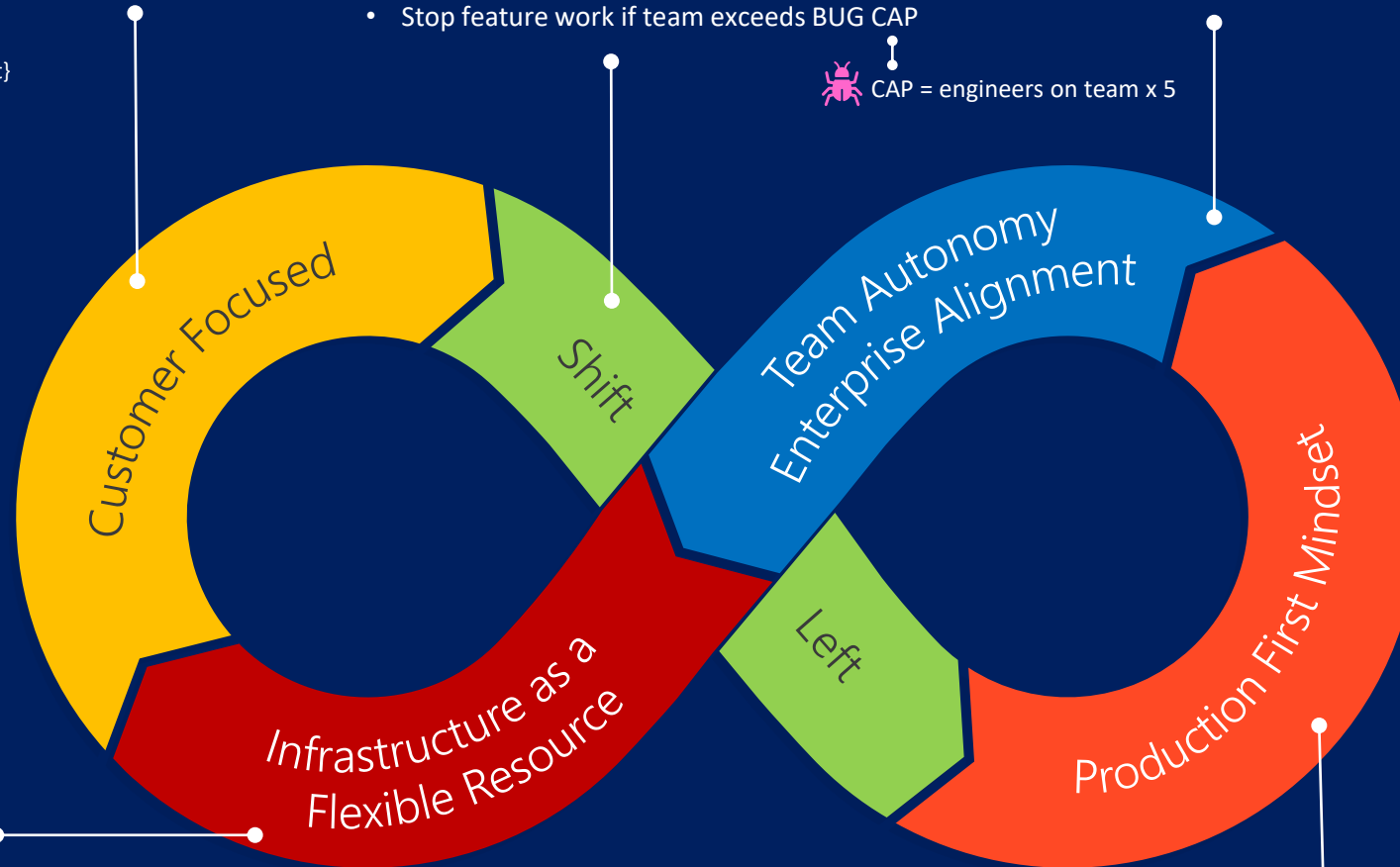
- Cloud and micro-services
- Containerization to raise redundant and scalability, and lower cost

- Shift left on testing and security
- Fast and reliable tests that run anywhere, by anyone, even in production
- Enforce reviews, validation scanning, and approvals with git Pull Requests.
- Stop feature work if team exceeds BUG CAP



CAP = engineers on team x 5

- Common cadence – heartbeat
- Feature team owns feature from idea into production
- Management owns WHAT + WHY we're building
- Engineering owns HOW we're building features
- Transparent collaboration



Engineering own the HOW



Be transparent
Visualize

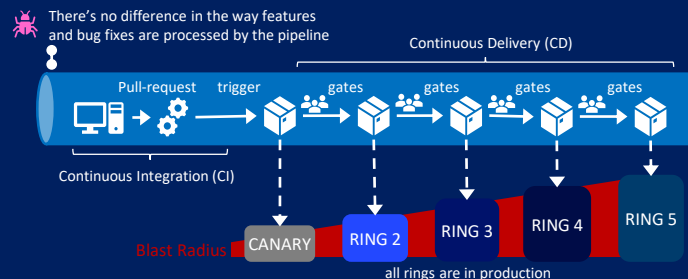


Sprint Plan



Sprint Retrospective

- Crisp 2-3min
- Keep it real!
- Sprint Highlights
- Sprint Video(s)
- Next Sprint Plan



- Remediate at root cause level with designated response individuals (DRI)
- Be transparent about issues, root cause, and resolution
- Fine tune alerts to be actionable
- No one-time fixes or commands – automate and version everything!
- Rings to limit blast radius and deploy releases progressively