* some trends in software development
  + decline in use of commercial software
  + focus on getting software into production
* challenge with becoming software-driven
  + lack of trust
  + oppressive technical debt
  + lack of automation
  + competing priorities
* Core DevOps Values
  + trust
  + empowerment
  + accountability
  + continuous improvement
  + data-driven decisions
  + customer empathy
* characteristics of a DevOps environment
  + optimized for throughput
  + clear view of entire deployment pipeline
  + customer-centric definition of "done"
  + small frequent software releases
  + defined feedback loops
  + automation-centric
  + focused on outcomes NOT activities
* changes you *may* see while adopting DevOps
  + philosophy changes
  + team changes
  + tool changes
  + satisfaction changes
* some core objections to DevOps
  + "We're already doing DevOps. We have a QA and release team."
    - DevOps means you have a cultural shift in developing and deploying software
  + "We don’t have the tech skills startups do."
    - it takes an investment
  + "Our compliance needs are too complicated for this."
    - DevOps actually makes things MORE secure
* Monday
  + org/team standup (team members, interested parties… 3 to 5 minutes)
    - transparency, throughput, delivery focus
    - **WHY?** orients everyone, draws attention to important information, coming together to remove blockers, facilitates continuous improvement
  + assign on-call engineer (rotating person on the application team)
    - develops empathy, focus on the customer
    - **WHY?** puts the focus on the customer and the service, encourages code instrumentation and improvement, helps other team members stay focused, record experiences to help the team
  + plan next sprint (whole application team)
    - focus on small batches and delivery pipeline
    - **WHY?** product owner maintains backlog, start meetings with retrospective, team decides amount of work in sprint, spring scope doesn’t change, teams always ship at the end of a sprint, sprint window should keep shrinking
  + triage new feature requests (product owner, stakeholders)
    - continual improvement, customer focus
    - **WHY?** review all software requests regularly, cross-functional participation, immediately adjust backlog priorities
  + merge code with the master branch (all engineers)
    - small batches, limiting your work-in-progress
    - **WHY?** continuous integration drives confidence, test coverage is key to trusting automation, fast feedback reduces wasted time later on, frequent testing means smaller debugging surface, aim for successful builds
* Tuesday
  + org/team standup (team members, interested parties… 3 to 5 minutes)
    - transparency, throughput, delivery focus
  + handle support tickets (on-call engineers, front-line support)
    - responsiveness and accountability
    - **WHY?** primary responsibility of on-call engineer that week, using same ticketing system as support personnel, may become group effort to resolve, sparks ideas for new features to propose
  + patching server clusters (engineers on the application team)
    - consistency, responsiveness
    - **WHY?** upgrades are required up and down the stack, infrastructure as code, multiple approaches possible, consistently applied to each environment
  + pair of infrastructure feature (engineers on the application team)
    - cross-training, collaboration
    - **WHY?** hire generalists with specialties, all configuration details in source control, test rigorously on a production-like environment
  + detect service interruption (support staff, on-call engineer)
    - responsiveness and accountability
    - **WHY?** upfront instrumentation pays off, use facts to figure out causation, communication is key, over-alerting causes fatigue
  + send status updates to executives (team leaders)
    - transparency and trust
    - **WHY?** DevOps is a cultural change and requires transparency, share core *business* metrics, build momentum through demonstrated progress
* Wednesday
  + org/team standup (team members, interested parties… 3 to 5 minutes)
    - transparency, throughput, delivery focus
  + onboard new engineers (application team)
    - cross-training and customer focus
    - **WHY?** learn from source control + deployment + pipeline + wiki, good opportunity to test deployment pipeline, pairing accelerates readiness
  + attend monthly operations review (application team leaders and executives)
    - customer focus and continuous improvement
    - **WHY?** remaining focused on customer metrics, leverage collective knowledge, improves situational awareness for executives, not a problem-solving session
  + conduct incident retrospective (on-call engineer, stakeholders)
    - continuous improvement and transparency
    - **WHY?** not about assigning blame, assemble timeline, use facts to review what happened, explain what worked and what didn’t, develop action plan and assign owners, put record into a discoverable place
  + collaborate across teams (application team leads and engineers)
    - remove bottlenecks and improve flow
    - **WHY?** no team should be entirely independent, think locally and act globally, create a system for quick collaboration and decision-making
* Thursday
  + org/team standup (team members, interested parties… 3 to 5 minutes)
    - transparency, throughput, delivery focus
  + write knowledge base articles + release notes (application team)
    - knowledge sharing and improve throughput
    - **WHY?** as velocity increases harder for users to track changes, content must be easy to create edit and find, change log broadcast engine are important
  + upgrade deployment pipeline (application team)
    - increased automation
    - **WHY?** build a better audit trail over time, first continuous integration and deployment may be lightweight, partner with groups like InfoSec and Compliance, cross-functional skill come in handy
  + right-size team rosters (executives and team leaders)
    - responsiveness
    - **WHY?** application teams are NOT fixed size, engineer count may swell during key periods, rotate operations and security expertise, offers maximum flexibility to respond to changing business
* Friday
  + org/team standup (team members, interested parties… 3 to 5 minutes)
    - transparency, throughput, delivery focus
  + package code for release (continuous integration process, on-call engineer)
    - fast flow
    - **WHY?** package is the result of a successful continuous integration pipeline, multiple way to package an application, packages should include ALL pieces needed to deploy, ensure packages can be recreated
  + deploy app update to production (deployment pipeline and on-call engineer)
    - last part of delivery pipeline
    - **WHY?** should be heavily practiced that it becomes boring, multiple techniques available to minimize downtime, have a telemetry in plan to measure impact of release
  + attend team lunch (everyone!)
    - team-building and sense of unity
    - **WHY?** build relationships within and across teams, these connections are important in a crisis or during retrospective, meetups should be planned by team and executives
  + cross-train teams (anyone in the org.)
    - continual learning and transparency
    - **WHY?** "show and tell" to train and educate, useful for front-line support engineers, spark ideas within other teams, share results of experiments with new technology