DevOps Habits @ Microsoft

DevOps is the union of people, process, and products to enable continuous delivery of value to our end users. - @donovanbrown

RELEASE FLOW BRANCHING STRUCTURE

bugfix   topic   topic

PR   PR   PR   PR

master

feature   topic

bugfix

M120   M121   Cherry Pick

Fix bugs in master first. This is true even if there's a critical live site incident.

In master, live in production, and collecting telemetry that examines the hypothesis which motivated the deployment

Definition of Done (DoD)

Build, measure, learn!

Site status is first priority.

Remediate live site incidents quickly, so that users are not affected.

Measure the customer value using telemetry. Validate your hypothesis.

Talk to and listen to your users! Face time is invaluable!

Constantly monitor production and fix anything that goes wrong immediately.

If bug count starts rising, stop development work, and fix the issues / bugs.

Run L0 & L1 in the pull request (PR) builds

L0 — Requires only built binaries, no dependencies
L1 — Adds ability to use data base and file system
L2 — Test a service via REST APIs
L3 — Full environment test end to end

Validated learning / Value improvement

Delivery Cycle

Gather feedback continuously and frequently

Deployment frequency

Ship smaller "rocks" (features), more frequently!

HABITS

- Evidence Gathered in Production
- Live
- Focus on Flow of Customer Value
- Rigorous Management of Technical Debt
- Team Autonomy and Enterprise Alignment
- Manage Infrastructure as a Flexible Resource
- Site Culture
- Hypothesis Driven Development

TEST PORTFOLIO

Completed Runs

Total tests	Failed tests	Pass percentage	Run duration
72751 (+72751)	0 (+0)	100% (+100%)	6m 41s (+6m 41s)

Passed (72751) New (0)
Failed (0) Existing (0)
Others (0)

Bug Cap

engineers on your team x 5 = limit

PLANNING

Leadership is responsible for driving the big picture

Sprint 3 weeks	Plan 3 sprints	Season 6 months	Scenario 18 months
1	3	40% 6	18

Teams are responsible for the detail

- We're continuously planning.
- A plan's not useful.
- Planning is useful.
- It's going through the thought process of planning where the value was.

Accelerated Agile
- Scrum / Feature teams
- 8-10 eng + PO / team
- Common backlog
- Aligned enterprise goals
- Continuous planning

Automate deployments, spin up more resources as needed, and harden your service.

Everything is evidence that we need in order to build up or draw against a set of hypothesis of beliefs.

HYPOTHESIS

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

ON / OFF

if (flag)
else

FEATURE FLAGS

Flags decouple release deployment and feature exposure, give run-time control down to the individual user, and enable hypothesis-driven development.

STAY IN SYNC

Be transparent
Visualize

3-week sprint

Week 1	Week 2	Week 3	Progressive Deployment

Sprint Plan

Sprint Retrospective
- Sprint Highlights
- Sprint Video(s)
- Next Sprint Plan

- Crisp 2-3min
- Keep it real!
- Real user experience

There's no difference in the way features and bug fixes are processed by the pipeline.

Continuous Delivery (CD)

Pull-request   trigger   approval   approval   approval

Continuous Integration (CI)

RING-BASED DEPPLOYMENTS

You can use either ring-based deployment or feature flags to implement the progressive exposure DevOps practice—they are symbiotic.

Blast Radius

CANARIES   RING 2   RING 3   RING 4   RING 5

ALL RINGS ARE IN PRODUCTION

With rings, we evaluate the impact, or "blast radius," through observation, testing, diagnosis of telemetry, and most importantly, user feedback. Rings make it possible to progressively deploy binary bits and have multiple production releases running in parallel.

Microsoft