

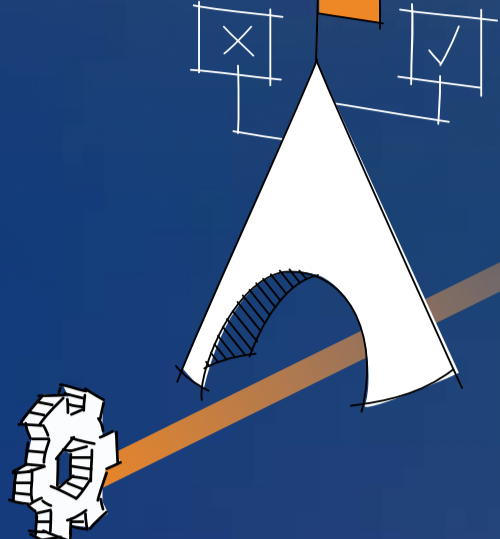
# The 4 Stages Of Feature Management Maturity

Confidence in software delivery is as mission-critical as the software delivery mechanism itself. While every engineering leader is under pressure to deliver software faster, many of them are not confident that they can do so safely. There's typically a tradeoff between velocity and risk – when one increases, the other decreases. **Feature management – often referred to as feature flags – mitigates this tradeoff.**

Several engineering leaders that indicated feature management is “mission critical,” to **Forrester** commented that:

*“Using Feature Management has completely changed the working atmosphere from one of high-stakes release processes rife with stress and error to one where feature release is part of everyday operations.”*

While many teams attempt to build this solution themselves, ultimately that limits their ability to implement and scale feature management as a key business process in line with CI/CD.



## Are you taking the right steps to ensure maturity and scale your feature management initiative?

The path begins with assessing where you are on the feature management maturity model, and then **investing in the right people, processes, and technology** to mature this mission-critical software delivery process.



## Start Your Journey



### PEOPLE

- Organization is new to feature management
- Only a small set of technical users is using feature flags
- Lower deployment stress and toil

### PROCESS

- Targeting specific users with application changes
- Production issues can be resolved with a kill switch instead of a rollback
- Separating feature release from production deploy

### TECHNOLOGY

- Organization is new to feature management
- Only a small set of technical users is using feature flags
- Lower deployment stress and toil

## IMPLEMENTING

## MANAGING

### PEOPLE

- Organization is new to feature management
- One or more teams using feature flags
- Non-technical users are empowered to manage flags
- Lower deployment stress and toil

### PROCESS

- Managing reporting of flag usage and feature adoption
- Handing off feature release to non-technical teams
- Creating “rules of engagement” for flag usage

### TECHNOLOGY

- Flag management UI
- Flag usage analytics
- Audit trails



### PEOPLE

- Feature management is well-understood across the organization
- Many or most teams are using feature flags
- Minimal deployment stress and toil

### PROCESS

- Creating standardized release processes
- Less manual management of individual flags
- Testing features live in production for A/B tests

### TECHNOLOGY

- Optimized to minimize downtime risk
- Integrated into developer stack
- Automated reporting

## AUTOMATING

## SCALING

### PEOPLE

- Feature management is a key business process
- All teams are using feature flags
- Developers can create, edit, and manage flags as code

### PROCESS

- Governance automatically enforced across all releases
- Predictable release process
- Minimal manual management of individual flags

### TECHNOLOGY

- Integrated system with CI/CD
- Flag lifecycle management
- Governance policy as code
- Automated progressive delivery pipelines
- GitOps with flags as code



## Feature Management Maturity

Ready to see how Harness **Feature Flags** can help you mature on this journey?

[Learn more](#)