# Continuous integration

* Less bugs get shipped to production as regressions are captured early by the automated tests.
* Building the release is easy as all integration issues have been solved early.
* Less context switching as developers are alerted as soon as they break the build and can work on fixing it before they move to another task.
* Testing costs are reduced drastically – your CI server can run hundreds of tests in the matter of seconds.
* QA team spends less time testing and can focus on significant improvements to the quality culture.

### What we need to do

* Team will need to write automated tests for each new feature, improvement or bug fix.
* We need a continuous integration server (jenkins) that can monitor the main repository and run the tests automatically for every new commits pushed.
* Developers need to merge their changes as often as possible, at least once a day.

# Continuous delivery

* The complexity of deploying software has been taken away. Team doesn't have to spend days preparing for a release anymore.
* We can release more often, thus accelerating the feedback loop with your customers.
* There is much less pressure on decisions for small changes, hence encouraging iterating faster.

### What we need to do

* We need a strong foundation in continuous integration and test suite needs to cover enough of your codebase.
* Deployments need to be automated. The trigger is still manual but once a deployment is started there shouldn't be a need for human intervention.
* Team will most likely need to embrace feature flags so that incomplete features do not affect customers in production.

# Continuous deployment

* We can develop faster as there's no need to pause development for releases. Deployments pipelines are triggered automatically for every change.
* Releases are less risky and easier to fix in case of problem as we deploy small batches of changes.
* Customers see a continuous stream of improvements, and quality increases every day, instead of every month, quarter or year.

### What we need to do

* Testing culture needs to be at its best. The quality of your test suite will determine the quality of releases.
* Documentation process will need to keep up with the pace of deployments.
* Feature flags become an inherent part of the process of releasing significant changes to make sure we can coordinate with other departments (Support, Marketing, and Production).