Continuous Integration

George Lesica - Wheeler Lab

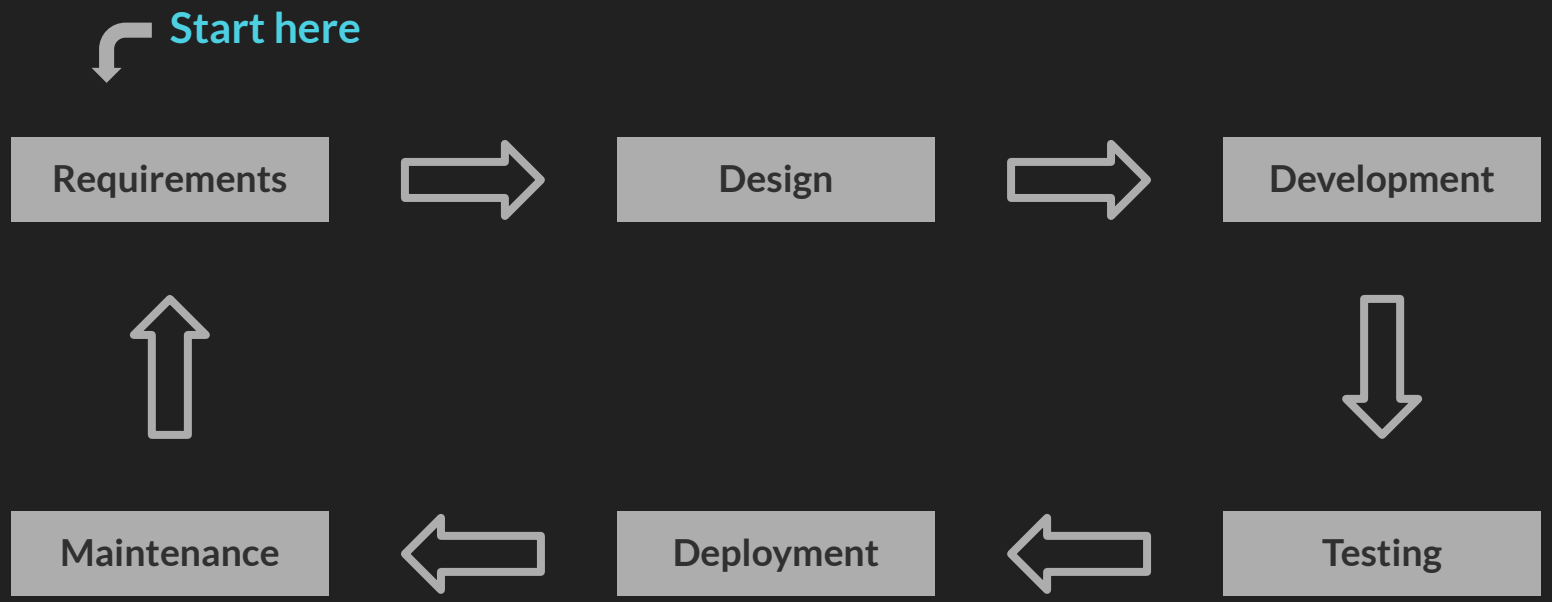
Continuous – constant,
all the time.

Integrate – to bring all the pieces together.

This usually means running quality checks and a build on every code change, but why?

Software development proceeds in stages…

Software Development Lifecycle



Let’s talk about Testing and Deployment.

Testing

Provides a quality signal to stakeholders

Allows changes to be made more quickly and reliably

Covers many different quality checks like formatting, benchmarking, and linting

Deployment

Requires that we deliver working code to stakeholders

Must happen quickly in the case of bug fixes

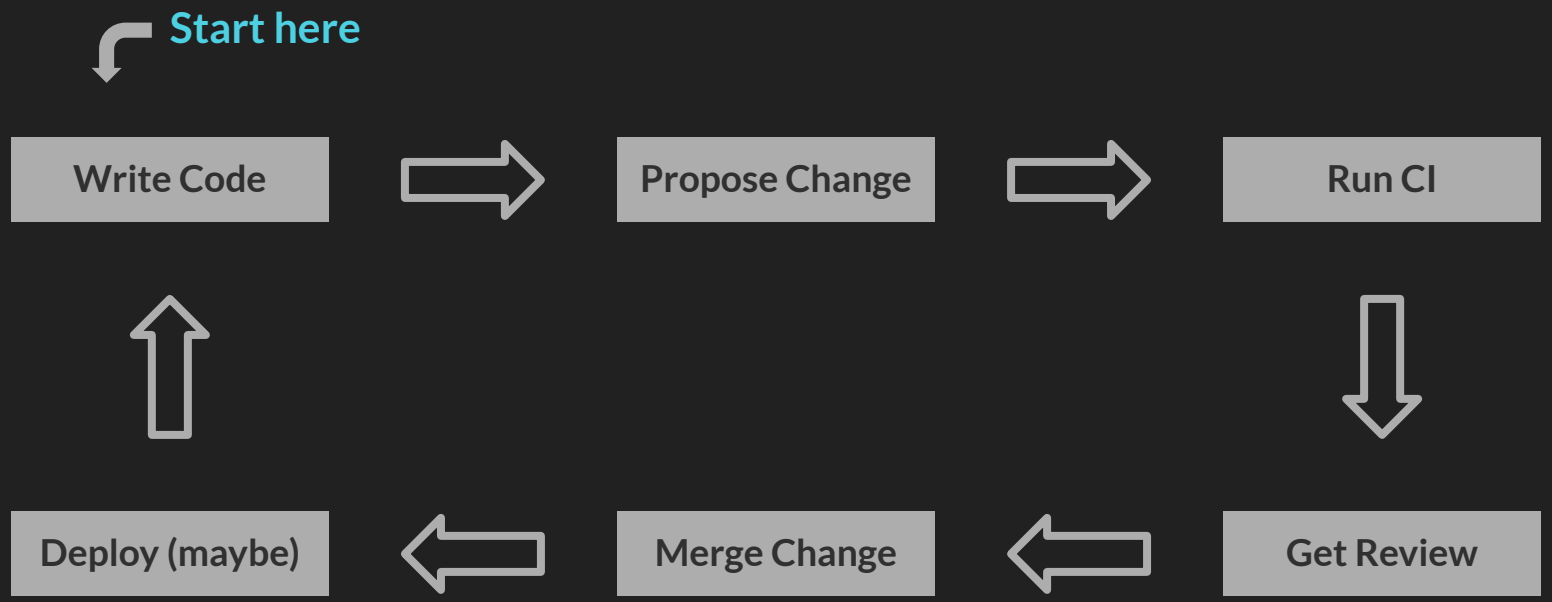
The process must be reliable

Agile increases the cadence and means we do these things over and over.

In order to deploy quickly, our code must be in working order at all times.

We use continuous integration (CI) to achieve this.

Developer Workflow with CI



Continuous Integration is like a sentry, guarding the codebase against unwise changes.

Changes are proposed through Pull Requests, they do not merge until CI has passed.

We use GitHub Actions to create CI pipelines.

GitHub Actions

Built into GitHub

Flexible (supports Docker)

Simple to configure

Production Configuration

For code that we expect people to use for Real Work™

Disable push to main / master

Require passing CI for PR merge

This is rolling out slowly to mature projects that have users (or that we want to have users)

Talk to George if you want help with the new workflow

Examples

PolyA – https://github.com/TravisWheelerLab/polyA

FastaVector – https://github.com/TravisWheelerLab/FastaVector