

# Use of Continuous Integration in BOINC Projects Development

# Software Project Housekeeping

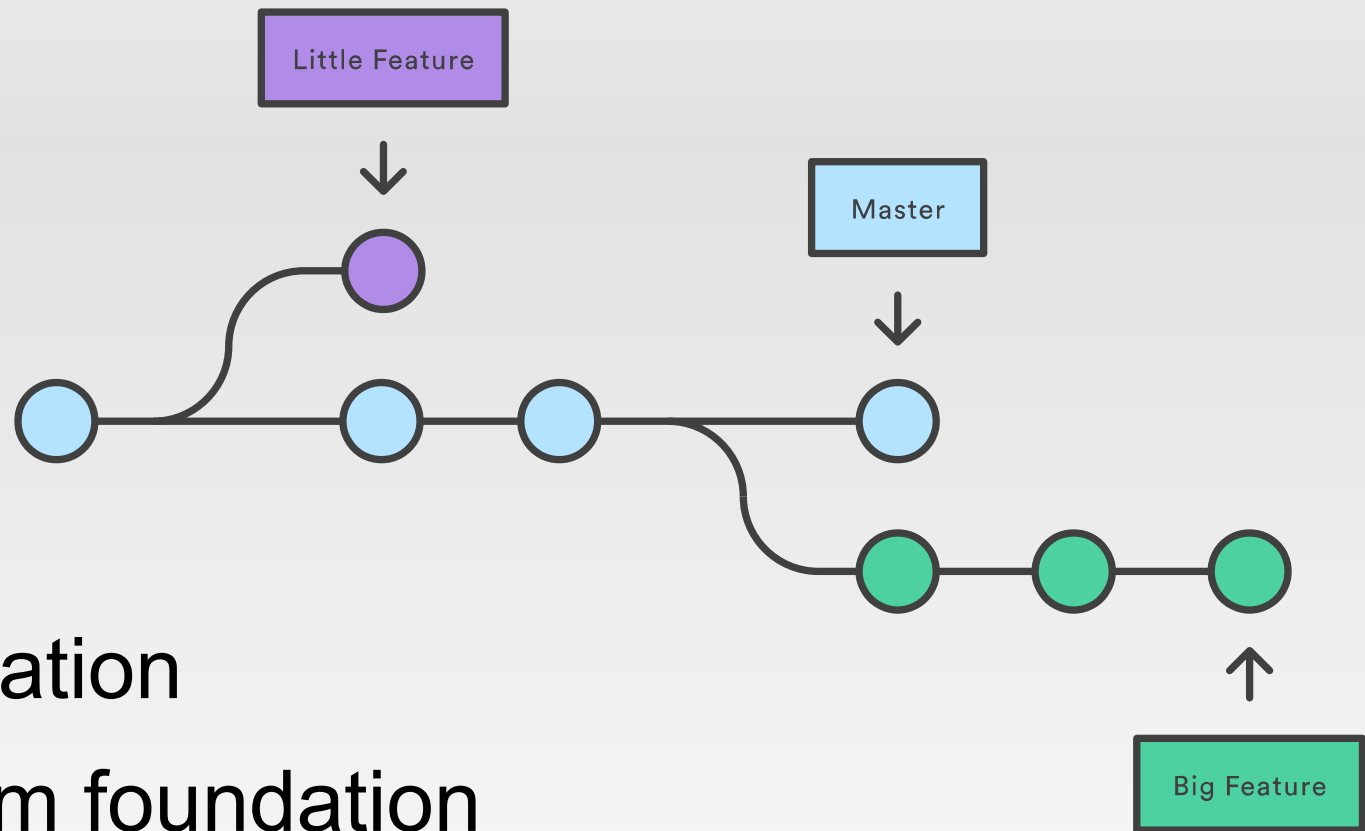
- Any sufficiently big software project should be set up in a way that allows **collaboration of multiple developers**
- Any project that is aimed at big user base should be **tested** prior to public release
- Every release should be **integrated** into company's infrastructure or **delivered** to users

# Continuous Integration

Software development **practice** that encourages developers **to integrate** new code changes into releases **as often as possible** by:

- Having a code **repository**
- Automating **builds**
- Automating **testing**
- Automating **deployment**

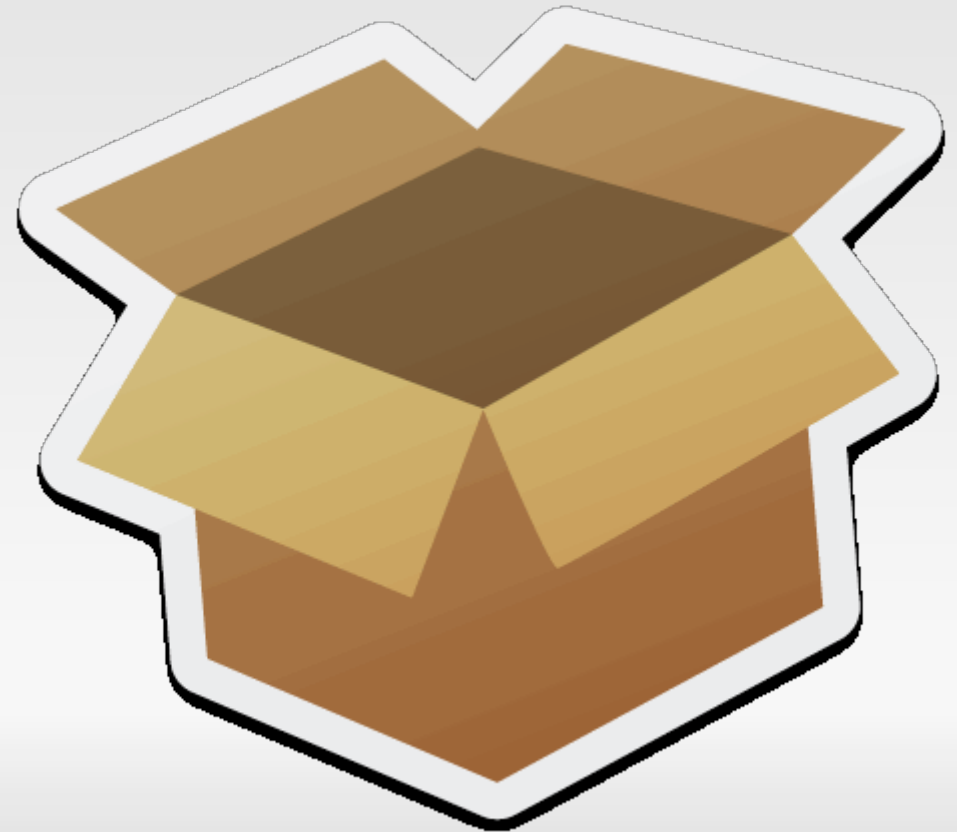
# Having a Code Repository



- History
- Collaboration
- CI system foundation

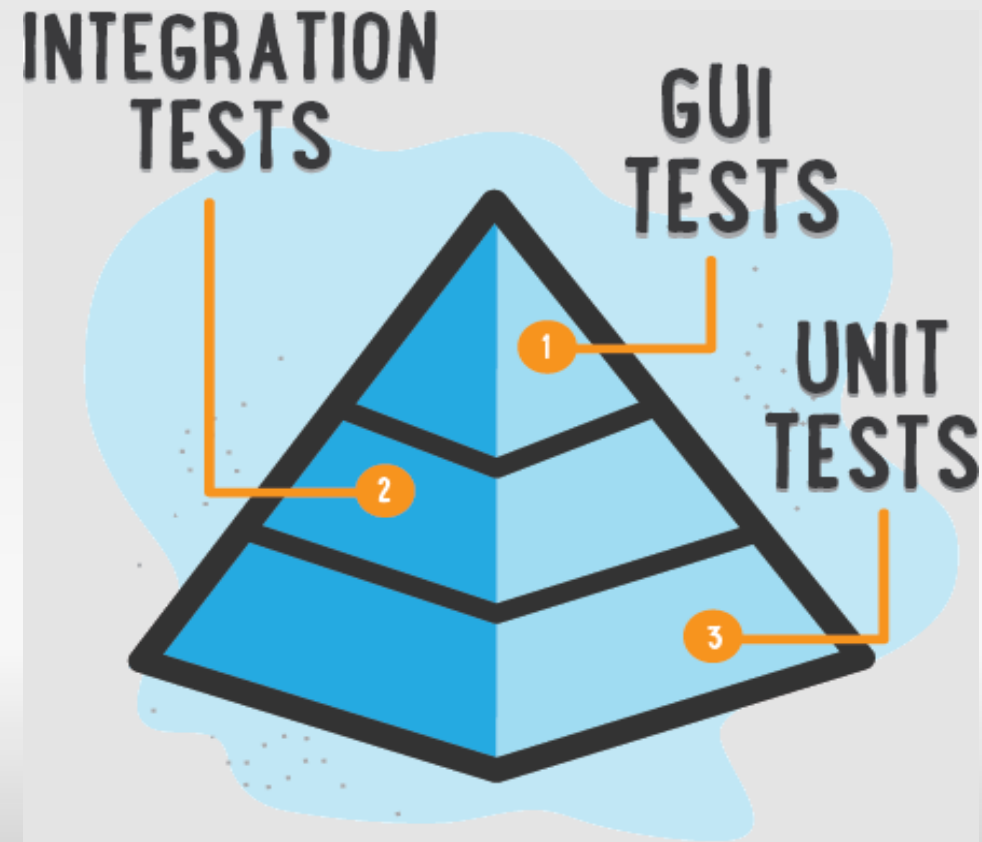
# Automating Builds

- Pull new code
- Satisfy build-time dependencies
- Perform compilation
- Save build artifacts



# Automating Testing

- Unit testing
  - Low-level
  - Language-specific
- Integration testing
  - High-level
  - Application-specific



# Automating Deployment

- **Application-specific** actions:
  - Deploy server software
  - Upload new design
  - Send e-mail to users
- **Linear** and **automatable**



# CI Pipeline Summary

- **Monitor** code repository for changes
- **Build** and **test** every commit, also **deploy** every commit to "public" branch of the repository
- **Fail fast** by prioritizing lighter steps
- **Notify** involved team members about **failures**



# CI Application in BOINC

## Choosing CI system:

- **Standalone** (Jenkins) or **integrated** with repo (GitLab)
- **Self-hosted** (both of above) or **cloud-based** (Travis CI, CircleCI)
- Form **simple** to really **complicated**

# Building BOINC Program

- **Clean** build environment
- **Isolate** independent projects
- Multiple **cross-platform** builds

# Testing BOINC Program

- Unit testing:
  - Prioritize **math-heavy** functions
- Integration testing:
  - Treat BOINC program as **server**
  - **Request-response** testing

# Deploying BOINC Program

- Delivery framework is **already present** in BOINC
- Invoke it from CI pipeline

# Conclusion

- Continuous Integration:
  - Automates **housekeeping**
  - Reduces **reaction time** on new bugs
  - Increases development **transparency**
- CI on BOINC:
  - Build process is essentially the same
  - Testing strategy is straightforward
  - Delivery framework is already implemented

Thank you for attention