Process and Incidents

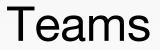


Date, time & occasion

Teams, not solo



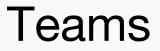
Date, time & occasion



Lower bus factor Personal responsibility



Date, time & occasion



Require code reviews Do code readings



Date, time & occasion

Pair Programming

Not instead of reviewing Helps newcomers on board



Date, time & occasion

Modularisation

Many small modules Internal libraries



Date, time & occasion

Modularisation

Internal interfaces Validate at interface edges



Date, time & occasion

One team, many modules

A team is **responsible** for their module Other teams send **patches** for review



Date, time & occasion

The Issue Tracker

All development goes through Issue Tracker All issues go as separate forks or branches



Date, time & occasion

The Issue Tracker

Jira (BitBucket), GitHub, Bugzilla, trac.....



Date, time & occasion

Review Issues

Bugs, Features, Description Treat them as prose / code



Date, time & occasion

Code Reviews

Require them (Mailing list, Gerrit, GitHub, etc.)



Date, time & occasion

Code Reviews

Unreviewed code is bad code Never merge bad code



Date, time & occasion

Track reviews

Flag tricky / interesting commits Use reviews as teaching aid



Date, time & occasion

Track reviews

Reviewed-by: Signed-off-by:



Date, time & occasion

Avoid "Velocity"

Prioritize Reviewing over Writing Reviewing is not Velocity



Date, time & occasion

Avoid singular work

Don't make one dev do all reviewing Don't make one group spend time reviewing



Date, time & occasion

Teach reviewing

Make management do it "Should be easy, it's just reading"



Date, time & occasion

Reward reviewing



Date, time & occasion

Reward teaching



Date, time & occasion

Don't accept lazy management

Managers shouldn't be allowed slack



Date, time & occasion

Involve management

Management should review issues



Date, time & occasion

Involve management

Management works overtime in Crunch



Date, time & occasion

Time based releases

Set schedule Cut features



Date, time & occasion

Major release

Every 6 / 9 months Not yearly, too big changes



Date, time & occasion

Minor release

Major + 3 weeks (bugfix) Major + 8 weeks (bugfix) Major + 12 weeks (bugfix)



Time based releases

Release every 5-8 weeks On schedule, mix features & bugs



Date, time & occasion

Time based releases

Track versions of modules Bump module version with Major release



Date, time & occasion

Use a release repo

Merge-only, release repository Tag all releases Sign tags



Date, time & occasion

Use a branching strategy

Tag releases Branches or Forks for Issues



Date, time & occasion

Branching Strategy

Plenty of well documented ones



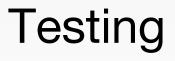
Date, time & occasion

Branching Strategy suggestion

master = next-stable branches/forks = development release = old-stable (tagged)



Date, time & occasion



Test teams are great Unittests are good Automation rocks



Date, time & occasion



Is a whole workshop alone



Date, time & occasion

Incident response

Have a process



Date, time & occasion

Incident response

- 1. Verify
- 2. Track
- 3. Document
- 4. Find
- 5. Fix
- 6. Release



Incident response

Needs to be practiced



Date, time & occasion

Use Incident response internally

Don't paper over internal finds



Date, time & occasion

Practice incident response

On security issues caught in review Use the response process



Date, time & occasion

Not a heavy process

Don't over do it Make sure many get training



Date, time & occasion

Track externals and respond

Libraries, OS, upgrades



Date, time & occasion

Have a security contact



Date, time & occasion

Don't shoot the messenger

Will ruin your PR Will impact your future



Date, time & occasion

Respond quickly

< 12 hours



Date, time & occasion

Keep reporter posted



Date, time & occasion

After a compromise

Different process, harder



Date, time & occasion

When? Why? What? How?



Date, time & occasion

Clone drives Dump logs



Date, time & occasion

All servers in UTC timezone? NTP running? Timeline wrt. Logs



Date, time & occasion

Backup integrity? Never restore



Date, time & occasion

New systems in place You have automation for this



Date, time & occasion

Find their way in before going live



Date, time & occasion

Assume credentials stolen Of all users with access Everywhere.



Date, time & occasion

Take your time, Panic won't help



Date, time & occasion

Audit trails Logs Timestamps



Date, time & occasion

That's a whole other course again. Talk to us if interested.



Date, time & occasion

Have a Process

Practice your process



Date, time & occasion