@therockertester

Shift Nowhere: a Testing Phase FTW

Dr Lee Hawkins

Who Am I?

Twitter:	@therockertester
Blog:	therockertester.wordpress.com
Books:	leanpub.com/anexplorationoftesters
	https://github.com/associationforsoftwaretesting/navigating

Template: Slides Carnival Ursula
http://www.slidescarnival.com/ursula-free-presentation-template/804

What Do I Mean by "A Testing Phase"?

A distinct set of test activities collected into a manageable phase of a project, e.g., the execution activities of a test level.
ISTQB Glossary

- ✗ A testing phase is a distinct period of testing effort, typically scheduled after the development phase is complete (signed off).
- ✗ This deliberate phase is found in all SDLCs, including waterfall and agile.

Why Fight the Industry Standard?

- ✗ A testing phase has always been a feature of software development lifecycles. There are phases for requirements/specification, design and development, so there must be a testing phase! It’s a best practice.
- ✗ Most software development teams still make use of a testing phase.
- ✗ But the sound and proven idea of a testing phase is currently under attack by the “shift left” and “shift right” movements.

> “Overall, the testing phase is an essential part of the software development process, as it helps ensure that the software is of high quality and fit for its intended purpose.”

Source: the ultimate arbiter of truth (aka ChatGPT)

Testing Too Early

- ✗ It doesn’t make sense to start testing until all the requirements have been coded.
- ✗ Trying to test earlier likely results in the need for retesting after further changes are made to the software under test. This is wasteful.
- ✗ The “Shift left” movement tends to ignore the downsides of testing too early.
- ✗ A dedicated testing phase help to prevent you from testing too early and so is also more efficient.

Testing Too Late

- ✗ Late changes to the software under test are a problem if there is no scheduled testing phase that only starts once all development is complete.
- ✗ The “Shift right” movement suggests that testing in production is a good thing, but there are many risks involved in doing so. Why take these unnecessary risks?
- ✗ A dedicated testing phase help to prevent you from testing too late, by scheduling testing at the optimal time, i.e. *after* development is complete but *before* release to production.

Gatekeeping

- ✗ Testers need to be independent arbiters of quality and a testing phase supports the required distance between other team members (e.g. developers) and testers.
- ✗ To adequately control quality, a testing phase is required. Only then can testing fulfil its key role as the gatekeeper of quality.
- ✗ A dedicated testing phase provides essential gatekeeping, ensures quality and results in meaningful sign-off on the software under test.

@therockertester

Testing phase FTW... Because

1. Industry standard & best practice
2. Prevents testing too early
3. Prevents testing too late
4. Enables essential gatekeeping