

Test Reporting in the Hallway

Moss Nye

Context

I've always worked for companies that develop software for other businesses, with a ***software as a service*** model.

I'm used to working with evolving teams and software, ***adding and supporting new capabilities*** over time.

When do you report?

When something has changed.

That is relevant to a stakeholder's interests.

The Medium is a Message

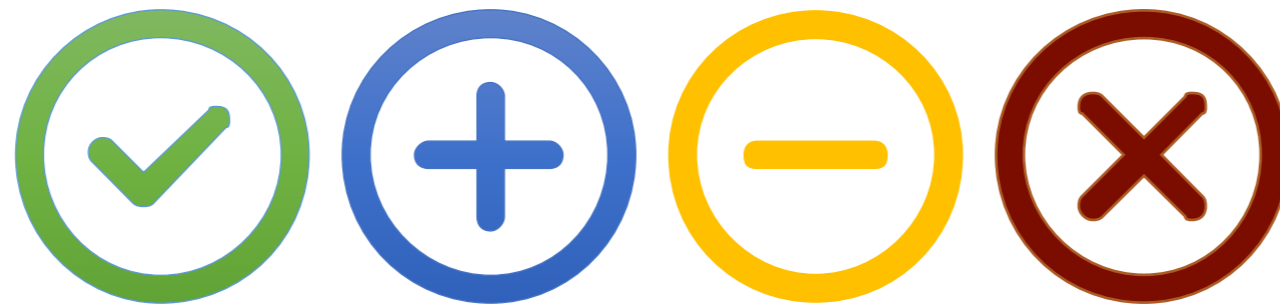


Copy of
Test Report
FINAL(1)
.docx.pdf

Summary

- ✔ State clearly the working capabilities of the software under test
- ✔ Celebrate the achievements of the testing & development team
- ⊕ Provide information and concise examples
- ⊖ Note any differences or deviations from initial plans
- ⊗ Raise key areas of concern

Recommendations

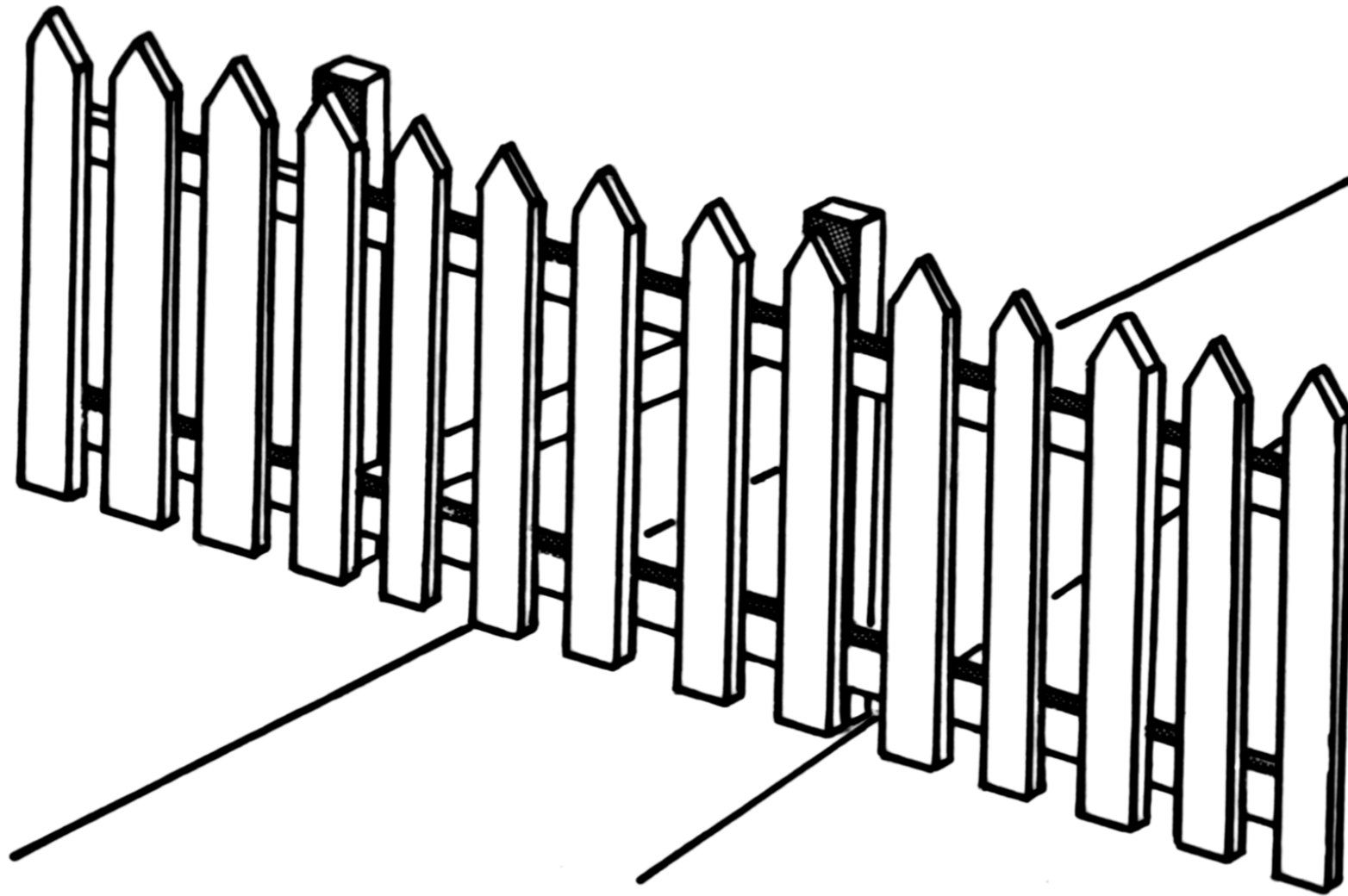


Read Feminist Philosophy

Recommendations

- ⊕ Take notes ***while you test***
- ⊕ The easier it is to discard your notes, the more useful they become.
- ⊕ Individual points for the summary will develop as you go

Existing Processes



What happened when I started adopting this

My test manager asked me what my process for testing a mobile application was.

About reporting I said...

“I finish all the checklists. Then I wait a day...”

What I could have said was...

**“After completing all the individual checklists,
I spend a day collating and cross-checking.
This helps me to...”**

✔ State Clearly

- ✔ Restate any planned functionality that is part of the **working software**
- ✔ Use your own words to do so
- ✔ Some words don't belong here. Saying something like "*Luckily we found the issue*" undercuts your skills.
- ✔ Mention team members **by name** for work they have done

Provide Information

“As uncertainty increases, the amount of information that must be processed by decision makers increases.”

– Jay R. Galbraith

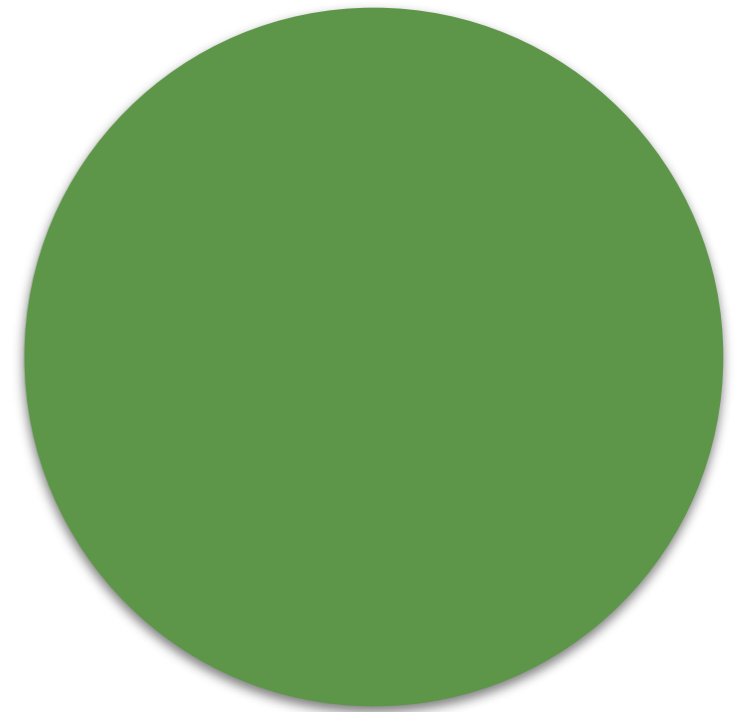
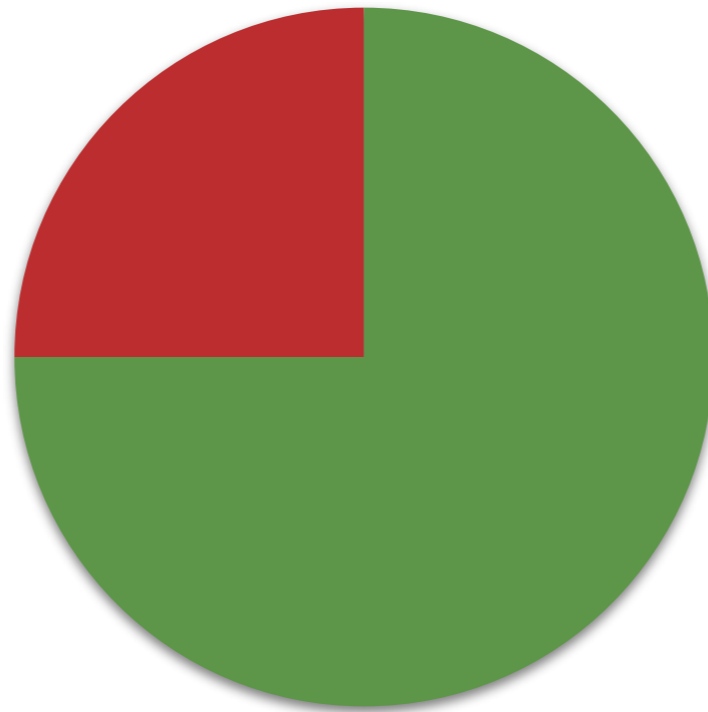
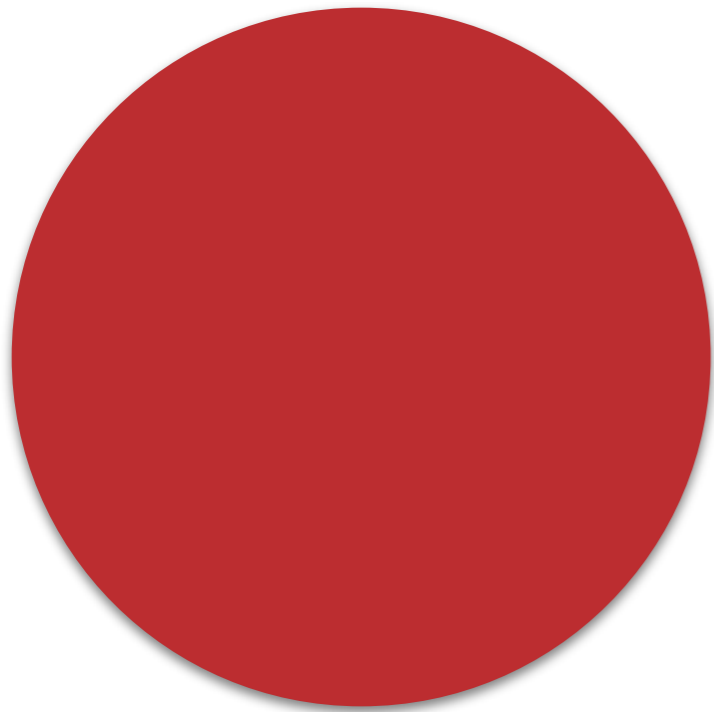
Organisation Design: An Information Processing View

⊕ Provide Information

- ⊕ You will be asked for **quantitative** metrics for **qualitative** concerns
- ⊕ There's a difference between metrics of software development and of working software
- ⊕ Stakeholders will say they need **X**, but they use **X** to derive **Y**. Find out what **Y** is and provide to that.
- ⊕ Document how usage of the software can be **observed**.

⊕ Provide Information

Friends don't let friends present with pie charts



Or count passed/failed test cases

"When a measure becomes a target, it ceases to be a good measure.

– Marilyn Strathern

⊖ Note Differences

- ⊖ I rarely see software delivered that is the same as originally planned, and I've never seen it in complex systems.
- ⊖ Testers are in a unique position to represent the **working software** as opposed to what was intended

⊗ Raise Concerns

- ⊗ An issue is a concern if **escalating** it would help
- ⊗ Escalate as an interruption when you are confident that the issue affects what the stakeholder cares about
- ⊗ Escalate asynchronously or through regular channels when you're sure it's a concern, but you're not sure how important it is
- ⊗ A rule of thumb is, will this **surprise** a stakeholder.

The Message isn't the Medium



Stakeholder Summary



“When you expose a problem you pose a problem. I have been thinking more about the problem of how you become the problem because you notice a problem.”

– Sara Ahmed. Feminist killjoy.