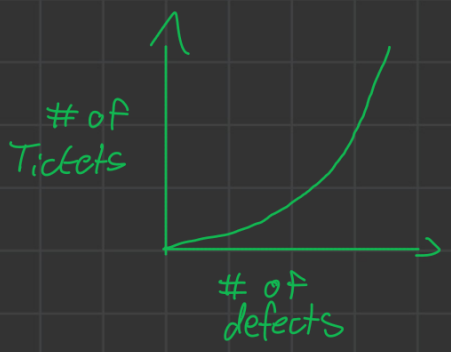
(1)



- domino effect: in communication, as much more parties are involved in fixing complex problems
- defect ping-pong: teams of developers passing defects tickets between each other, no one assumes full responsibility

- the more groove the software buy, the more people (exponentially) will file reports
- more people needed for product support as more users try to contact for support → costly
- user satisfaction is lowered → lower demand

(2)

For:
- ressource or time limitation
- developers know their code in depth
  ↳ can implement unit testing easier
  ↳ regression testing
- early bug fixing
- dev testing own code results in less code

Against:
- testers are faster and more skilled at writing tests
- different perspective
  ↳ devs in tunnel mode, code blind
- missing non-functional aspects
- testers also test for unintended use of software
  ↳ click a button twice quickly for example
- "works on my machine"

(- Social Conflict: Devs don't like to be criticized for bugs within their dev team)

- feedback from testers valuable for dev

③

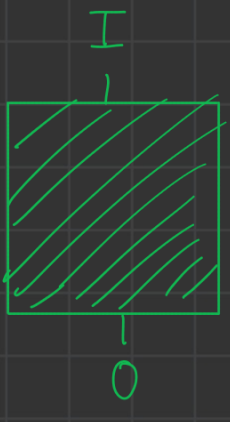
Equivalence class	Input Values	Test case
[content size same]	{("null", " null "), ("", "<a>"), ...}	Same size
[content size not same]	{("null", "nullo"), (" ", " a ") ...}	Not the same size

↳ same for html & text comparison -> equal & not equal
↳ other possibilities:
User registration:
- Login username valid/invalid

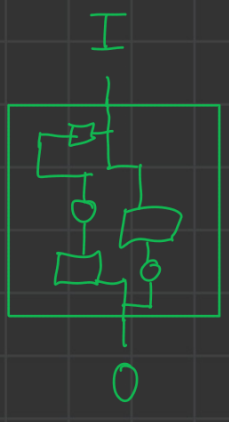
④ Regression testing:
- Tests of all Guis are being veran after any change to the codebase
=> Bugs/Errors can be tracked down precisley and quickly

⑤

Black Box testing



White box testing



- no access to source code
- dont care about
- test software functionality without considering internal logic
=> only for validating sw functionality

- know source code and logic
- test software functionality while considering the internal logic
=> unit-tests
=> better for finding defects