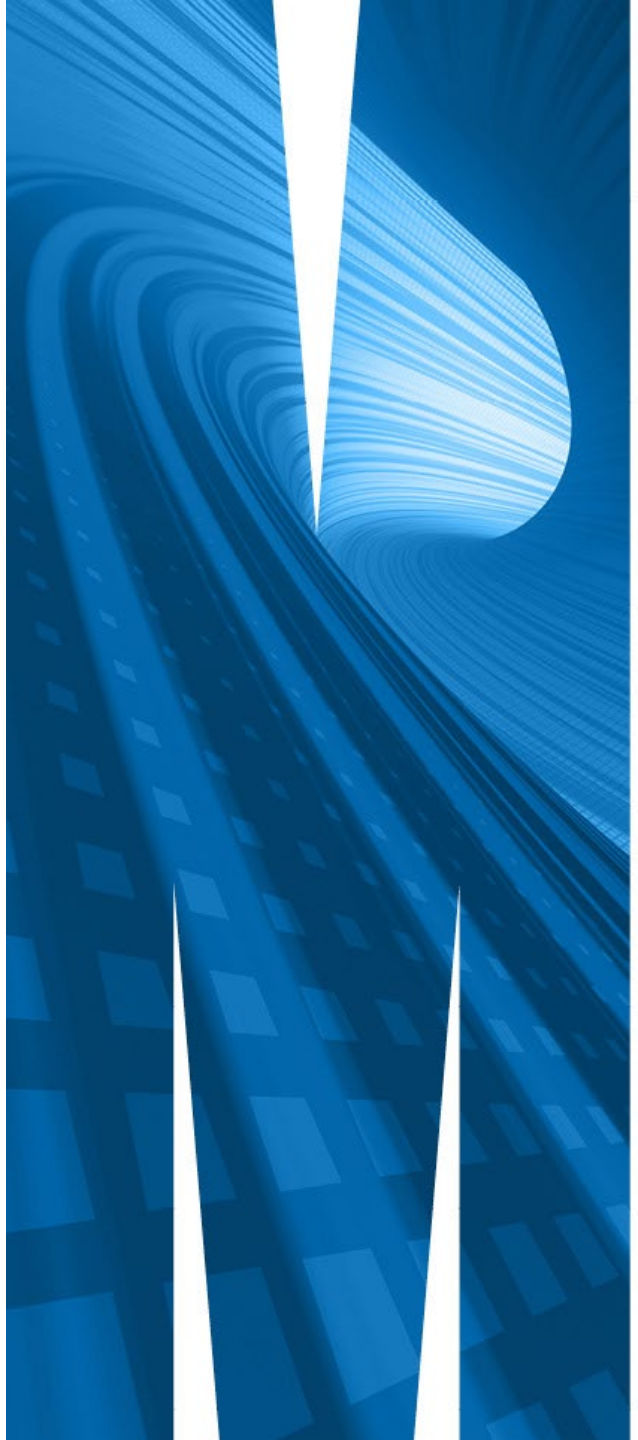


FIT2107-Software Quality & Testing

Lecture 11 – Software Quality Metrics

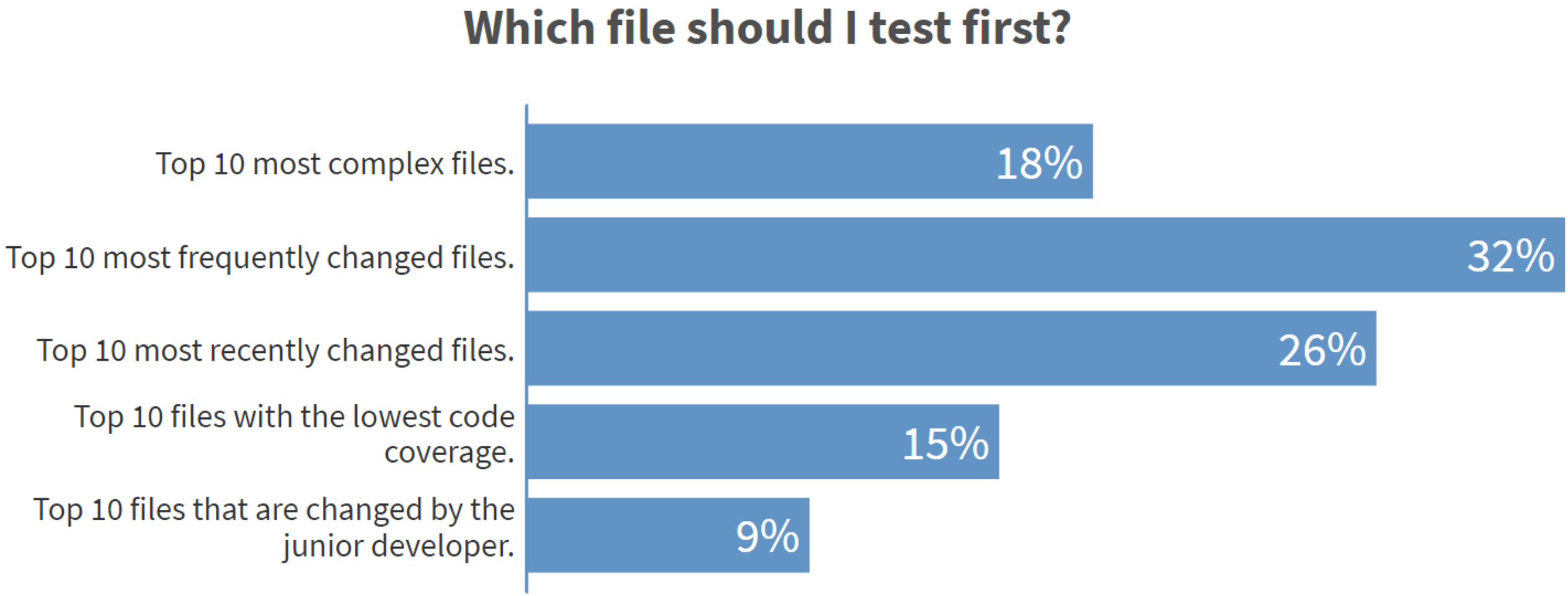
27th October 2020

Dr Najam Nazar

Announcement

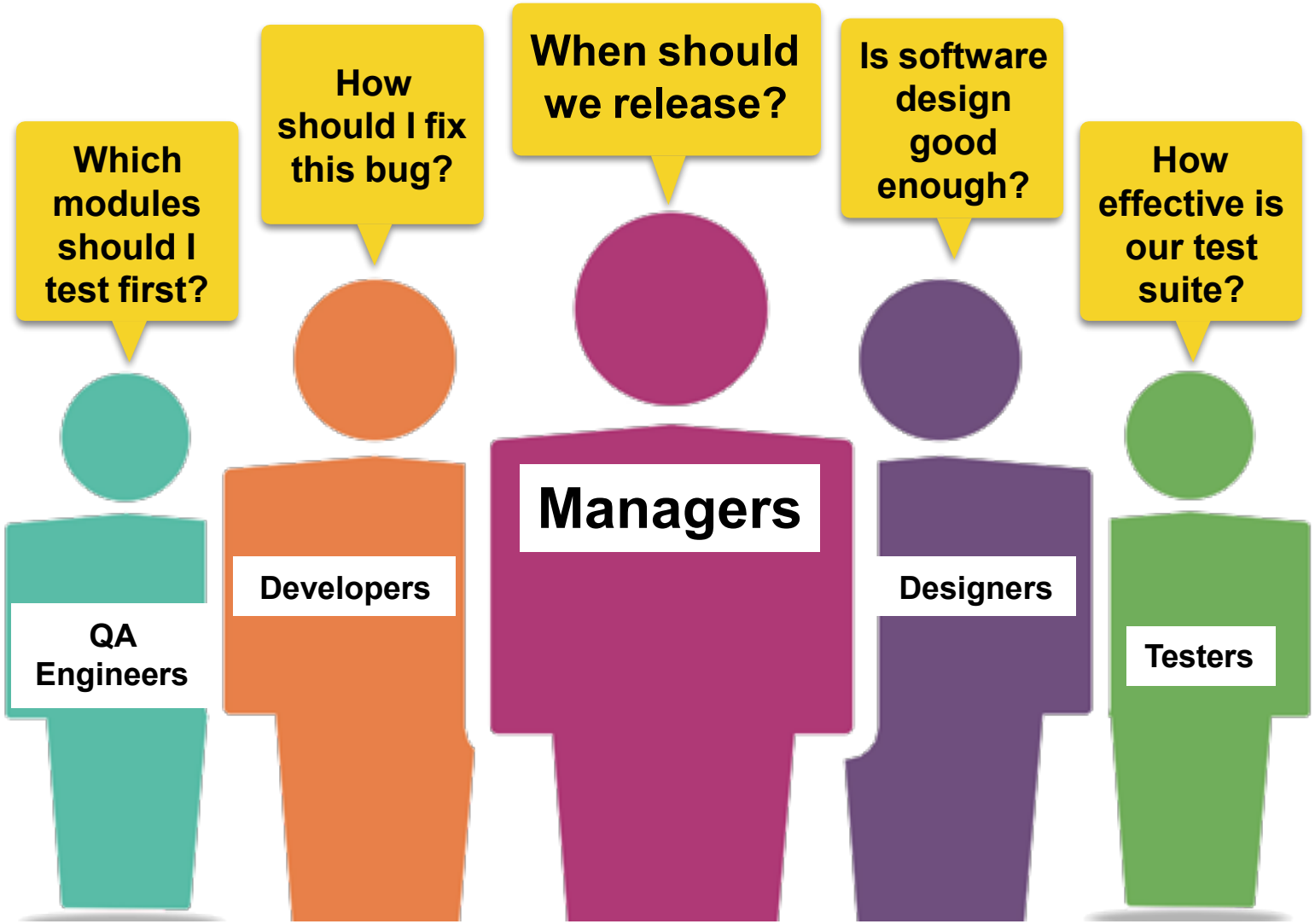
- Mock Exam is ready
- I will share the link on forums some time this week.

Poll Results



Decision Making

- Software are complex in nature.
- Throughout the SDLC, we need to make complex and critical decisions.
- Decision based on evidence.



Decision Making

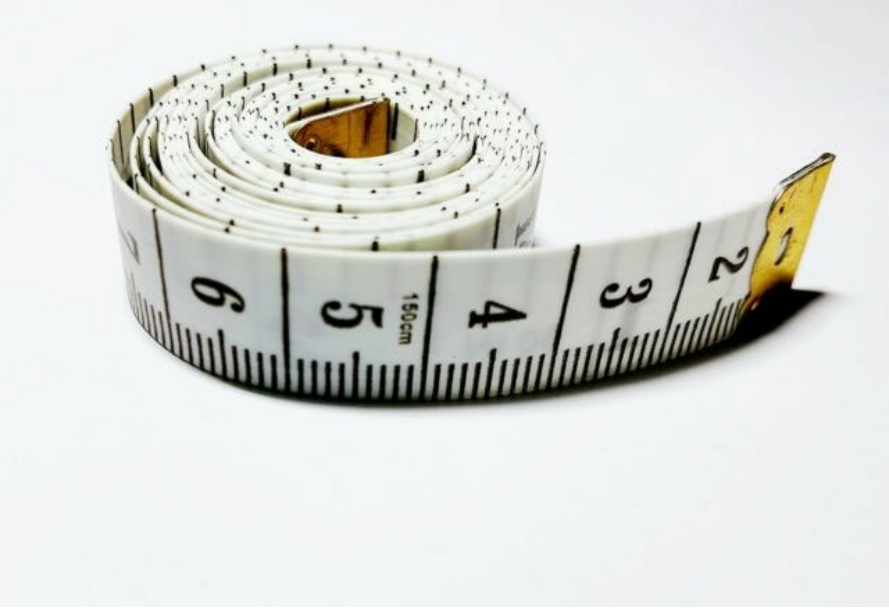
- It is important to collect data, measure it on certain factors and make decisions.
- Collect evidences.

Hiring an expert comes at high cost

Your gut feeling is too random and risky

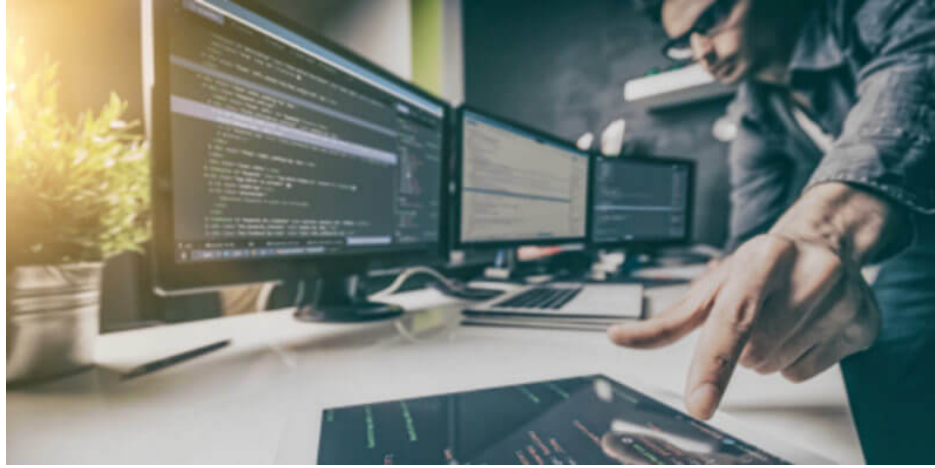
Software Metrics

Software Metrics



- Metrics refers to standards for measurements.
  - Second is standard measure for time
  - Metre is a standard measure of length
- Similarly software are measured too.
- Many matrices are developed to measure the software on different standards.
- PhD in software metrics.
- We only focus on quality metrics in this unit.

Purpose of Metrics



- Project Monitoring – Check the status
- Project Controlling – Corrective actions.
- Plan new projects
- Measure and analyse results
  - The profit of testing
  - The cost of testing
  - The quality of testing
  - The quality of the product
- Predictions about product before we build it.
- Assess product or process to decide what to do next.

Selecting the right metrics

- What is the purpose of the collected data?
  - What kinds of questions can they answer?
  - Who will use the data?
  - How is the data used?
- When and who needs the data?
  - Which forms and tools are used to collect the data?
  - Who will collect them?
  - Who will analyse the data?
  - Who have access to the data?

What quality aspects can we measure?

- Can we measure:
  - Functionality?
  - Reliability?
  - Usability?
  - Maintainability?
  - Portability?
  - Efficiency?
- …we can measure them all…
- But some are harder than others to measure validly.
- What else can we measure?
  - Tests?
  - Yes, we can…

Testing Metrics

Line of Code (LOC)

- Pretty easy to count.
- Should filter out comment lines.
- Lines per method/class a pretty good predictor of likelihood of bugs!
- Long methods and big classes are buggy…

Testing Metrics

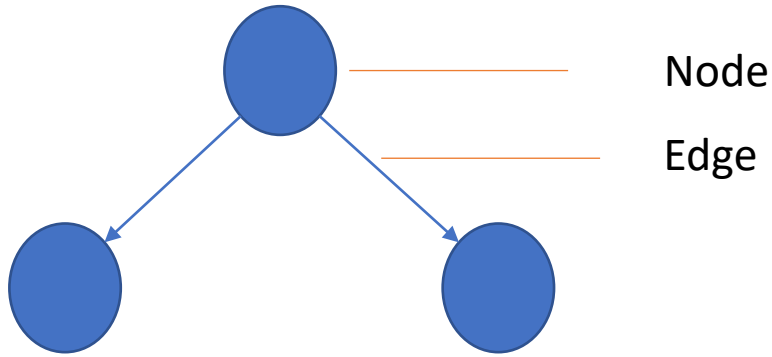
Code Coverage

- Coverage
  - Coverage.py -> statement and branch coverages
  - Higher coverage more faults can be identified.
- Can we say?
  - "x% coverage is sufficient to give you reliability"
  - No we can’t.

Testing Metrics

McCabe Complexity

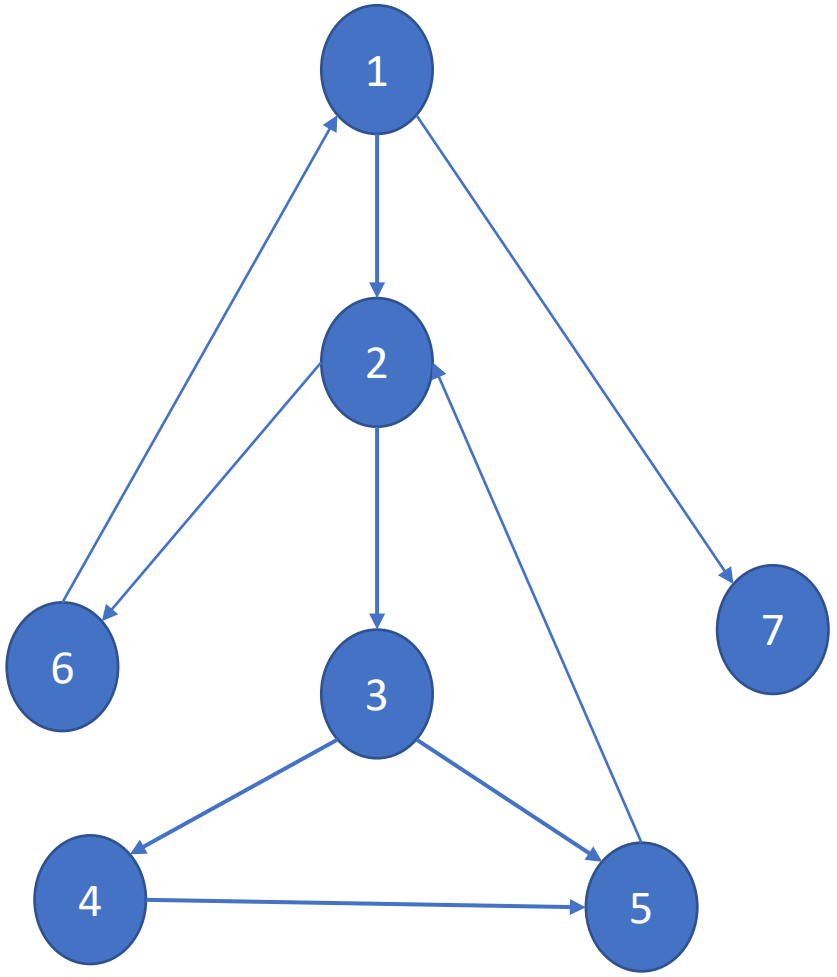
- Measures the complexity of a software program.
- A quantitative measure of independent paths in the source code of a software program.
- Measure?
  - CFG
  - Functions, modules, methods or classes within a software program
- Calculate?
  - V(G) = E - N + 2, where
    - E is the number of edges in CFG, and
    - N is the number of nodes in CFG



Testing Metrics

McCabe Complexity

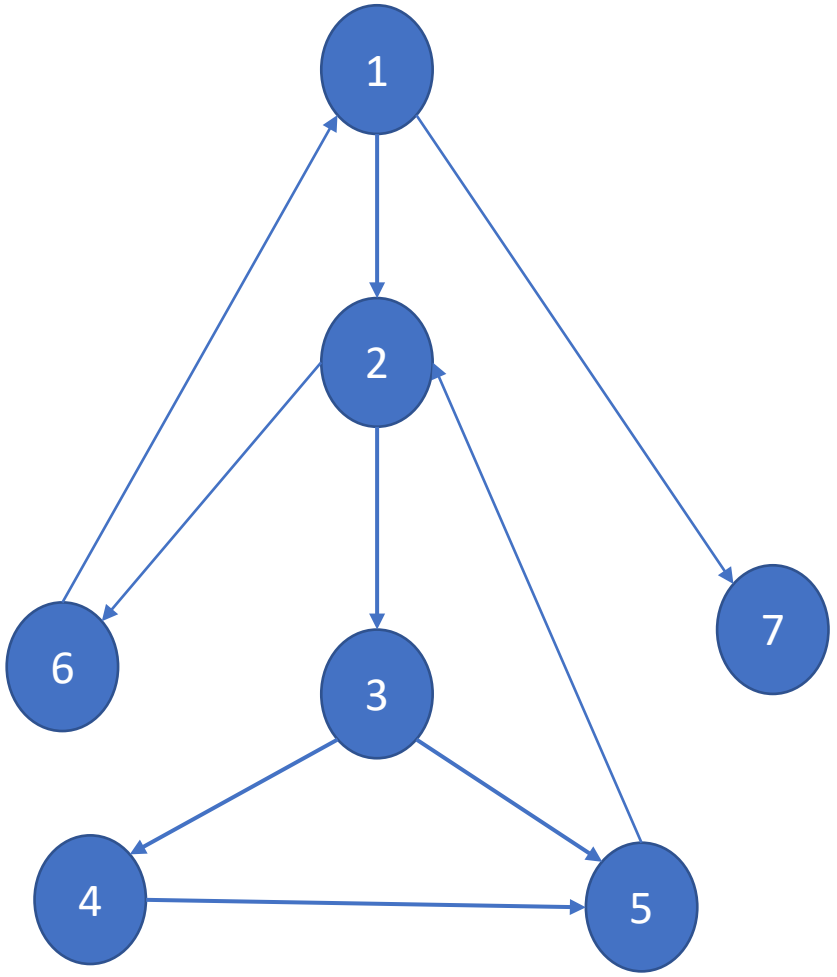
- How many edges we have?
  - 9
- How many nodes we have?
  - 7
- So what's the value?
  - E-N+2
  - 9-7+2
  - 4



Testing Metrics

McCabe Complexity

- How many condition nodes?
  - 3
- So what's the value?
  - P + 1
  - 3 + 1
  - 4
- What execution paths we have?
  - 1, 7
  - 1, 2, 6, 1, 7
  - 1, 2, 3, 4, 5, 2, 6, 1, 7
  - 1, 2, 3, 5, 2, 6, 1, 7



Rule of thumb >10 is hard to debug/test/maintain

Testing Metrics

Halstead Complexity

- Set of metrics computed from different features of source code.
- Measurable properties of the code and their relationship.
  - η1 = number of distinct operators.
  - η2= number of distinct operands.
  - N1= total number of operators.
  - N2= total number of operands.
- $V = N\log_2^{\eta}$
- $D= (\eta1/2) * (N2/\eta2)$
- ---

Object Oriented Metrics

Chidamber and Kemerer

- Proposed six OO design metrics.
  - Weighted Methods Per Class - WMC
  - Depth of Inheritance Tree - DIT
  - Number of Children - NOC
  - Coupling Between Objects - CBO
  - Response For a Class - RFC
  - Lack of Cohesion in Methods – LCOM
- Many of these can be calculated from a class diagram.

Object Oriented Metrics

Chidamber and Kemerer

Weighted Method per Class

- The sum of the complexities of the methods.

$$WMC = \sum_{i=1}^{n} c_i$$

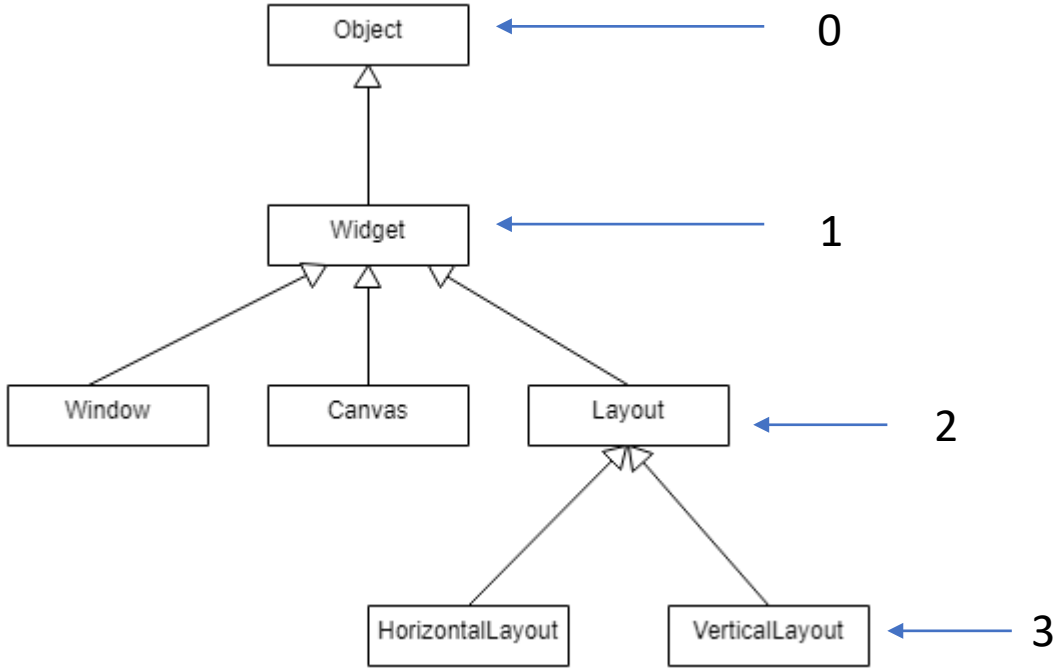
- The number of methods and the complexity of methods involved is a predictor of how much time and effort is required to develop and maintain the class.
- Reusability

Object Oriented Metrics

Chidamber and Kemerer

Depth of Inheritance Tree

- The maximum length between a node and the root node in a class hierarchy.
  - DIT0 -> Object
  - DIT1 -> Widget
  - DIT -> Window, Canvas and Layout
  - DIT3 -> HorizontalLayout and VerticalLayout
- High DIT means high complexity, difficult to maintain.

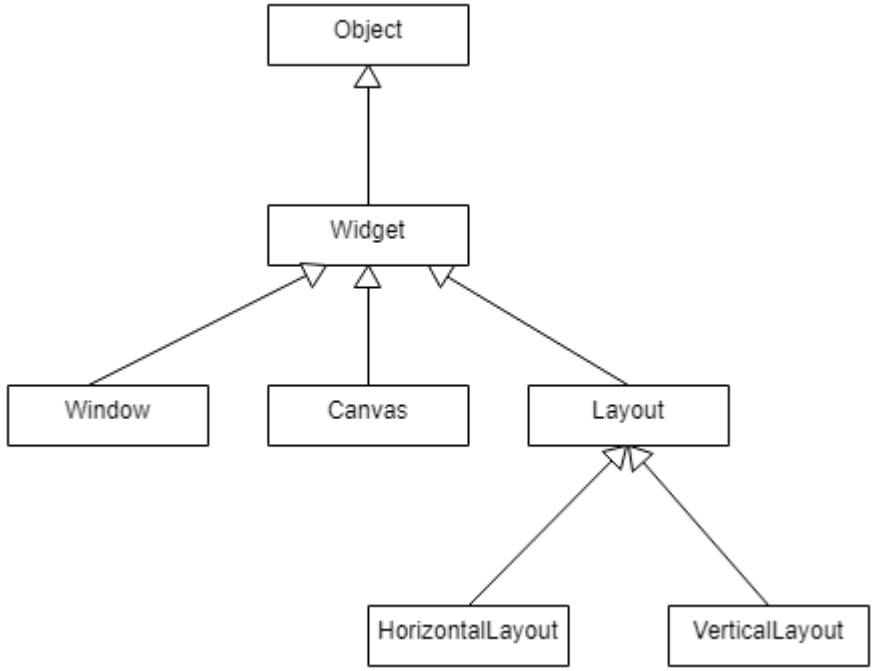


Object Oriented Metrics

Chidamber and Kemerer

Number of Children

- Number of classes inherited
  - Object -> 1 Child
  - Widget -> 3 children
  - Layout -> 2 children
- High NOC means good reuse but bad abstraction.



Object Oriented Metrics

Chidamber and Kemerer

Coupling between objects

- Number of classes that are coupled to a particular class i.e. where the methods of one class call the methods or access the variables of the other.
- Interaction between methods.
- Increased coupling increases interclass dependencies,
    - less modular
    - less reuse.
    - Difficult to maintain

Object Oriented Metrics

Chidamber and Kemerer

Response For a Class

- All the methods in the class and all the methods that are called by methods in that class.
- If a large number of methods can be invoked in response to a message, the testing and debugging of the class becomes more complicated since it requires a greater level of understanding required on the part of the tester.

Object Oriented Metrics

Chidamber and Kemerer

Lack of Cohesion in Methods

- Methods relate to each other.
- High cohesion low coupling.
- A low LCOM value suggests the class is more cohesive and is viewed as better.
    - Cohesiveness of methods within a class is desirable, since it promotes encapsulation.
    - Lack of cohesion implies classes should probably be split into two or more subclasses.
    - Low cohesion increases complexity, thereby increasing the likelihood of errors during the development process.

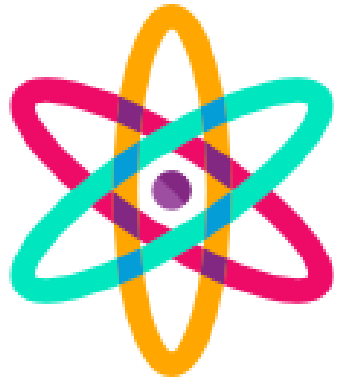
Fault Prediction

- A key quality property that would be nice to be able to predict is functional correctness.
- Not only do you want your code to be functionally correct, but if you have modules that are predicted to be buggy, they are the ones that you'll want to test most intensively!
- Machine learning model.
- Train model and predict faults.
- Big data.

Fault Prediction

Predicting future software defects so practitioners can effectively optimize limited resources

Explaining what makes a software fail so managers can develop the most effective improvement plans



Building empirical-grounded theories of software quality

Testing Maintainability

- After functional correctness other important factor we look for is maintainability.
  - Extend the code
  - Reuse code
- Do developer's perceptions of maintainability relate to actual quantifiable maintenance properties - in a business context, does it relate to time/cost to maintain?
- Does matrix created for other languages say Java, C fits for the language you use say Python.
- Tricky- depends on many factors.

Taylorism

- Scientific management
  - Measure everything.
  - Develop the “optimal method” through measurement.
  - Train your workers to follow method exactly.
  - Reward them for the desired outcome.

Code Review Metrics

- Measure the activity
- Code Review Coverage
  - The proportion of changes that have been reviewed in the past
  - The proportion of code that has been reviewed in the past
- Code Review Participation
  - #Self-Approved: The proportion of changes to a file that are only approved for integration by the original author.
- Code Review Ownership
  - The number of developers who have reviewed code changes to the file.
  - The degree of review ownership.
- Pull Requests.
- Merge Requests.

THAT'S ALL FOR THIS UNIT

Next Week

- Exam Discussion.
- Revision.
- Don’t forget to attempt the mock exam.
    - Link will be shared in the forums.


QUESTIONS