CS 301
Software Engineering
Module - 32

Eswaran Narasimhan

At the end of the session, you will be able to:

❑ Understand the basics of Software Quality

❑ Differentiate Software Quality Control and Software Quality Assurance

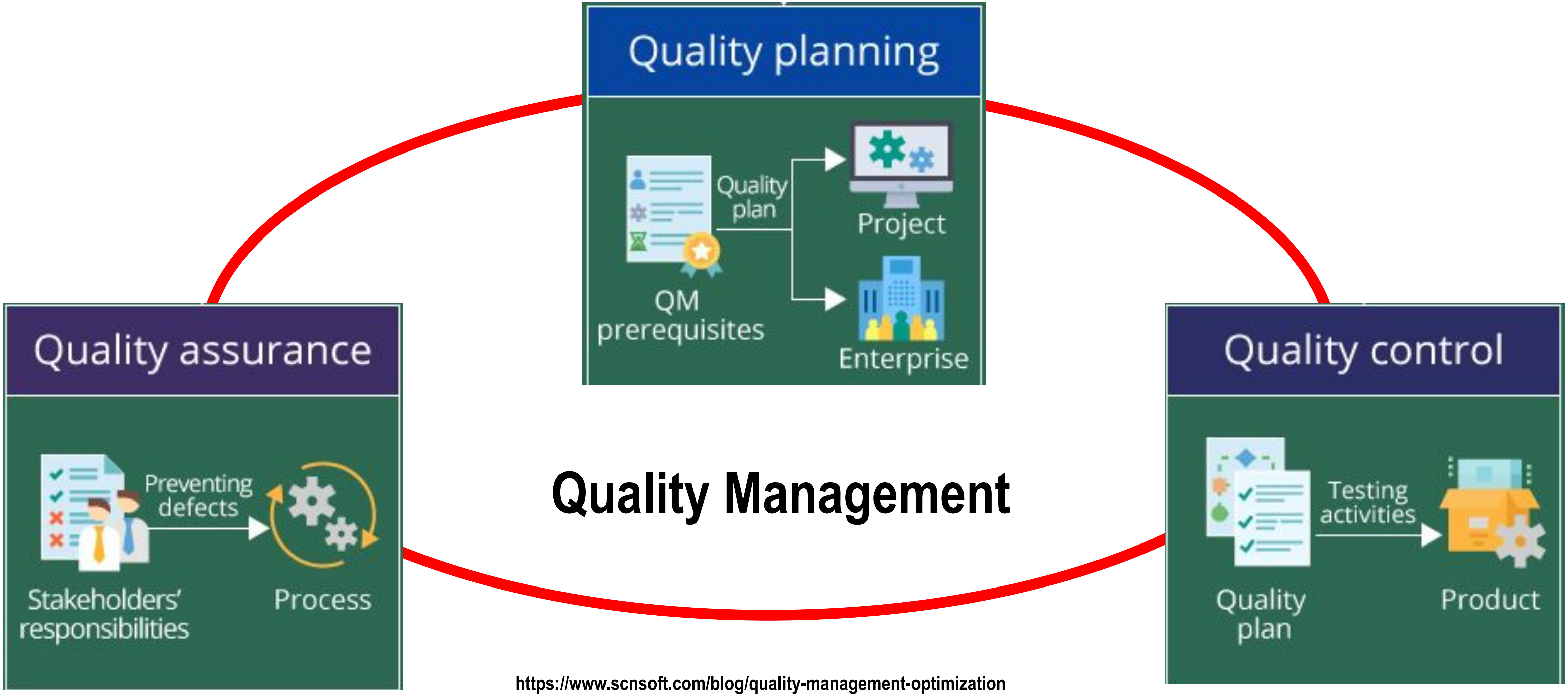
❑ Understand the use of metrics for Software Quality

❑ Understand Software Quality Control Processes

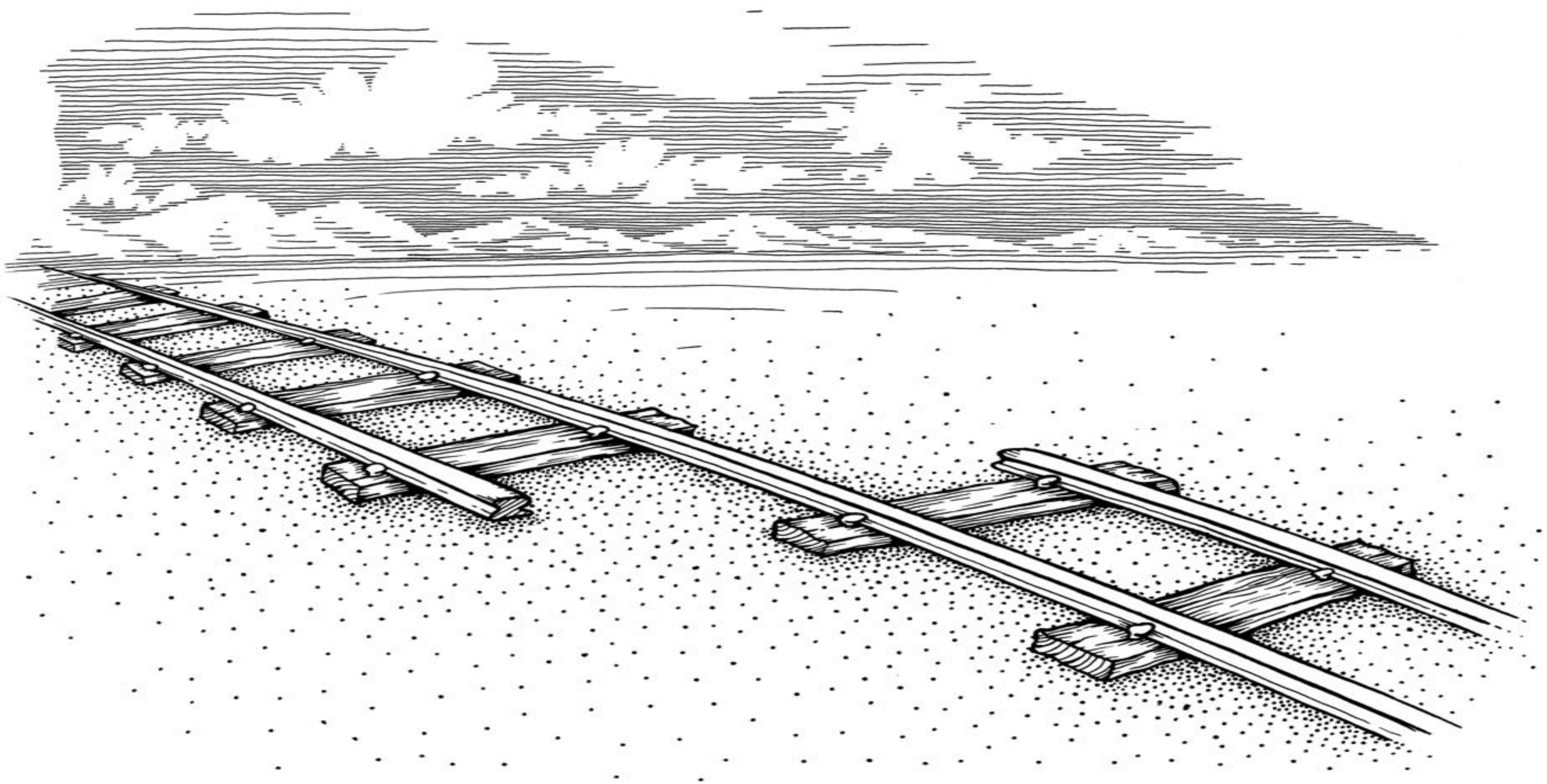
❑ Integrating metrics within software quality processes

❑ Understand software reliability and metrics

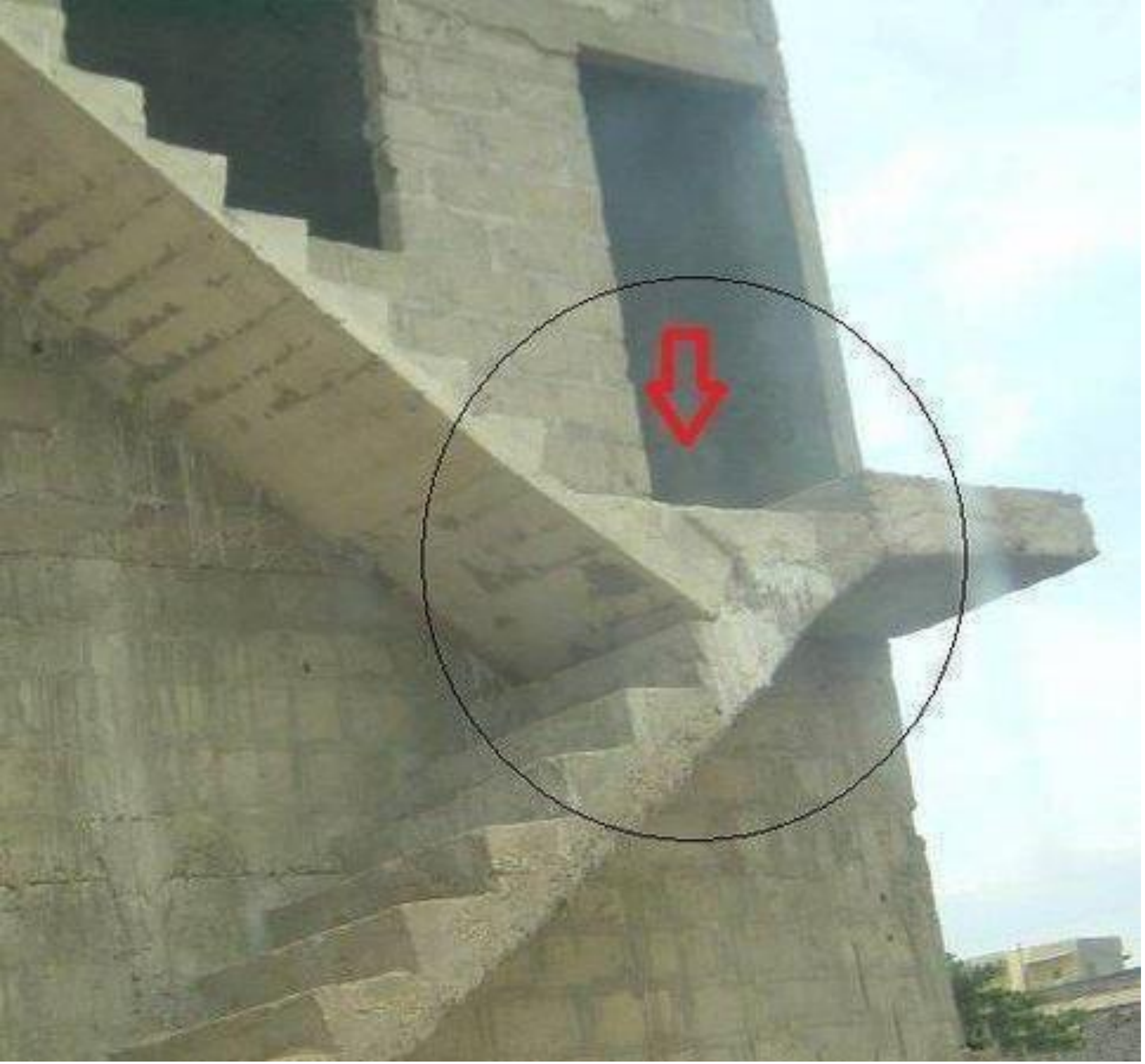
Software Quality Management



https://www.scnsoft.com/blog/quality-management-optimization

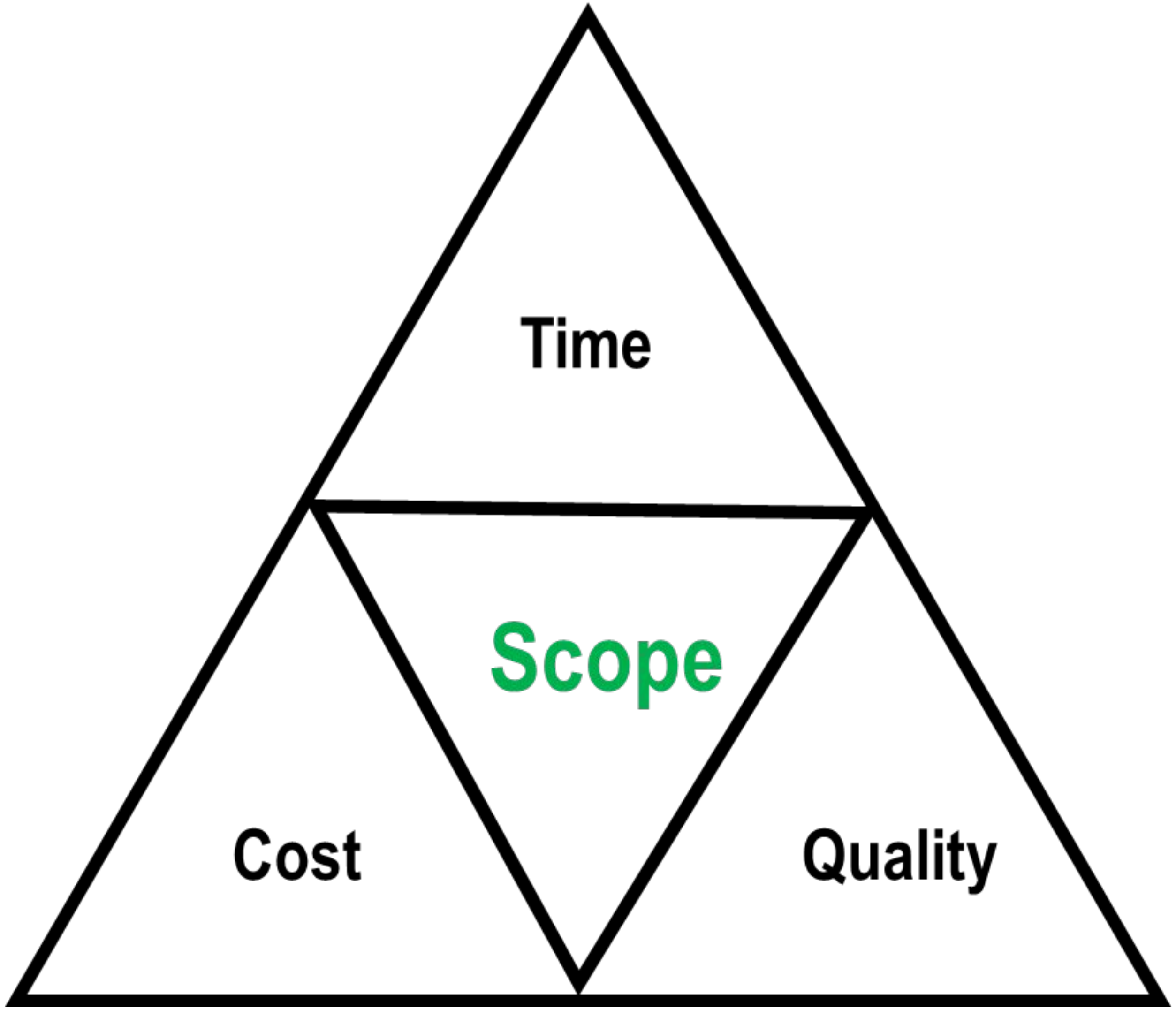


https://www.pritchettcartoons.com/wrong_rail.htm



https://www.pinterest.com/pin/463659724109611331/

Process Planning


Time

Scope

Cost

Quality

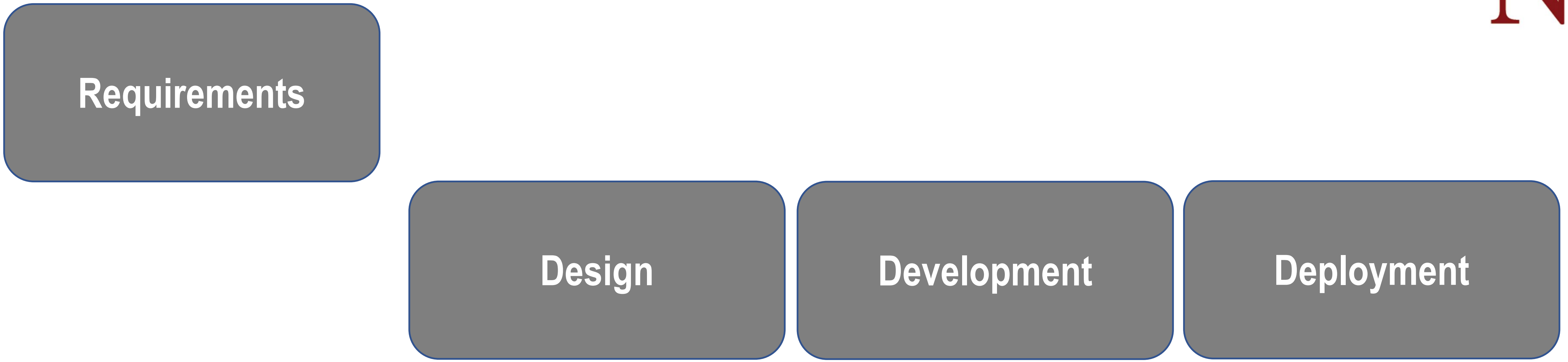
□ Scope coverage
□ As per requirements
□ As per legal standards

❑ No Overrun in time
❑ Within effort planned
❑ Within cost budget
❑ Within acceptability quality, both FR and NFR

Software Quality Assurance

- Providing assurance that quality will be achieved
- Aims to prevent the defect
- It is a verification process
- It does not involve executing the program
- It's a Proactive and Preventive initiative
- It is the procedure to create the deliverables
- Across the SDLC
- It defines standards and methodologies
- It is performed before Quality Control
- It requires the involvement of the whole team
- Use Statistical Process Control concepts

Metrics for Software Quality Assurance



Requirements

Design

Development

Deployment

Metrics for Software Quality

Requirements

- ❑ Requirements List
- ❑ NFR List
- ❑ Special requirements

Design

- ❑ Design Elements
- ❑ Layers and artifacts

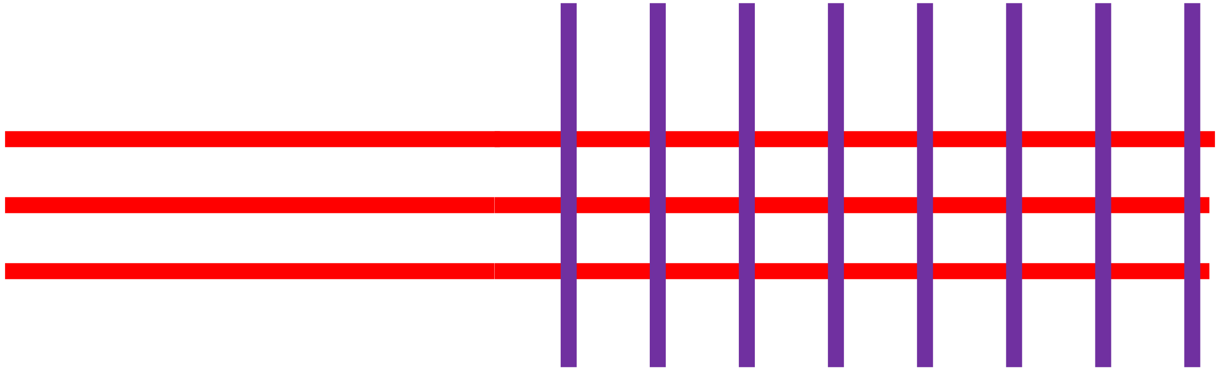
Metrics for Software Quality

Requirements

- □ Requirements List
- □ NFR List
- □ Special requirements

- □ Design Elements
- □ Layers and artifacts

Design



Metrics for Software Quality

Requirements

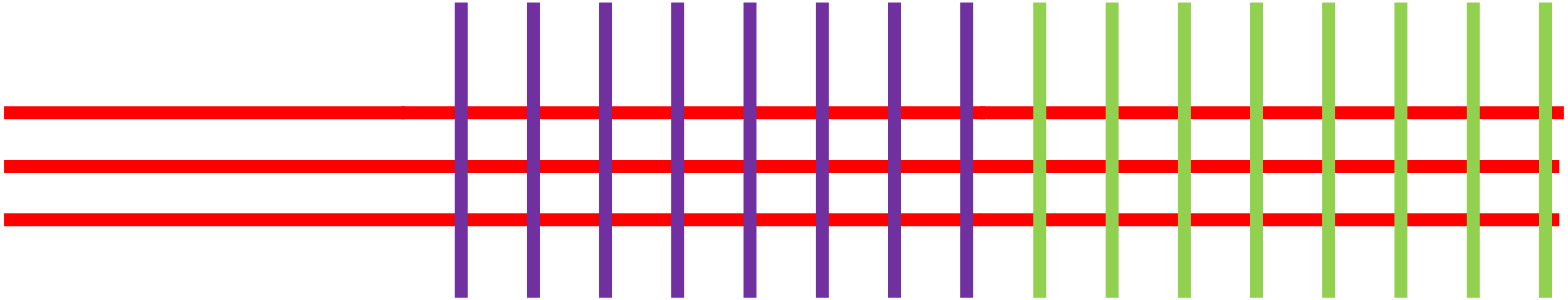
- ❑ Requirements List
- ❑ NFR List
- ❑ Special requirements

Design

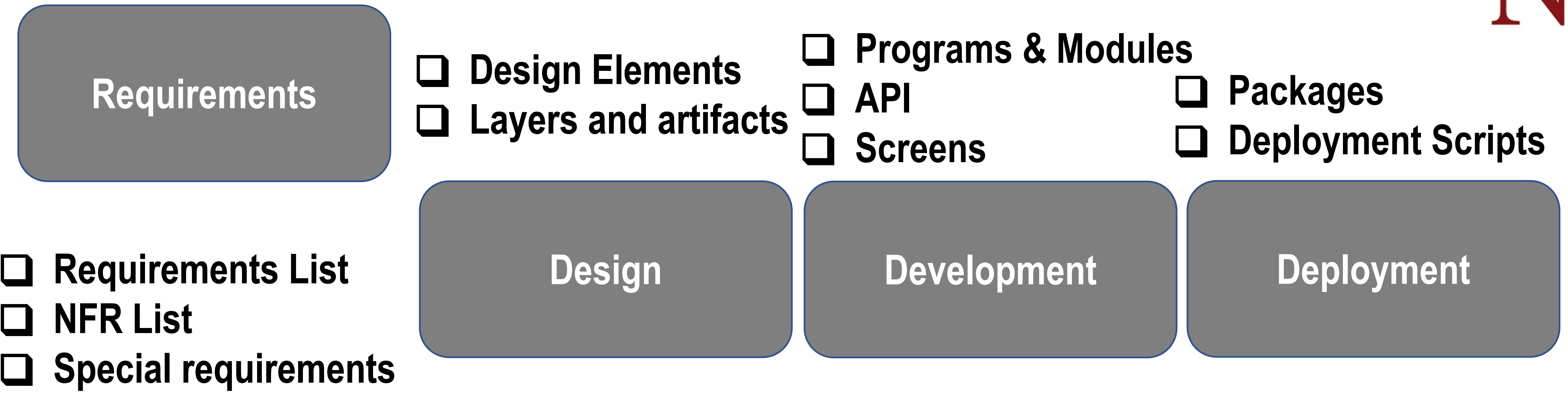
- ❑ Design Elements
- ❑ Layers and artifacts

Development

- ❑ Programs & Modules
- ❑ API
- ❑ Screens



Metrics for Software Quality



Requirements

- □ Requirements List
- □ NFR List
- □ Special requirements

Design

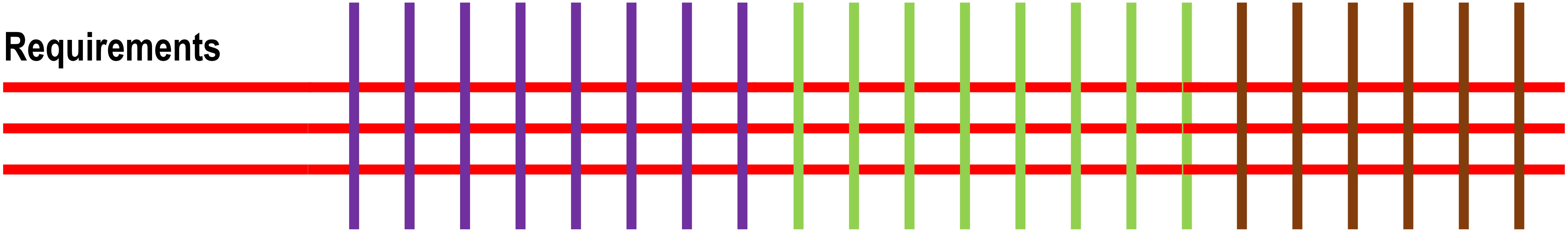
- □ Design Elements
- □ Layers and artifacts

Development

- □ Programs & Modules
- □ API
- □ Screens

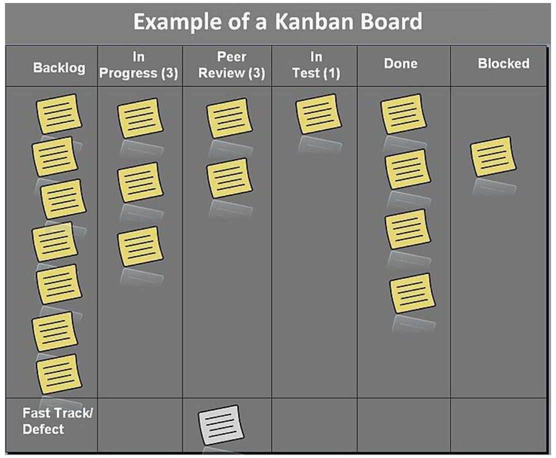
Deployment

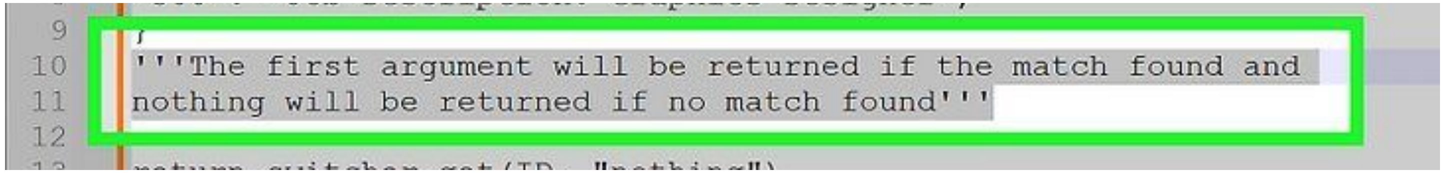
- □ Packages
- □ Deployment Scripts



Other practices for SQA

- ❑ Check in – Check out Processes
- ❑ Updating the Kanban Board
- ❑ Code Comments in programs
- ❑ Checklist maintenance for
  - ▯ Screen Completeness
  - ▯ Program completeness
  - ▯ Function calls and Parameter passing
  - ▯ Running tools such as lint





```
'''The first argument will be returned if the match found and
nothing will be returned if no match found'''
```

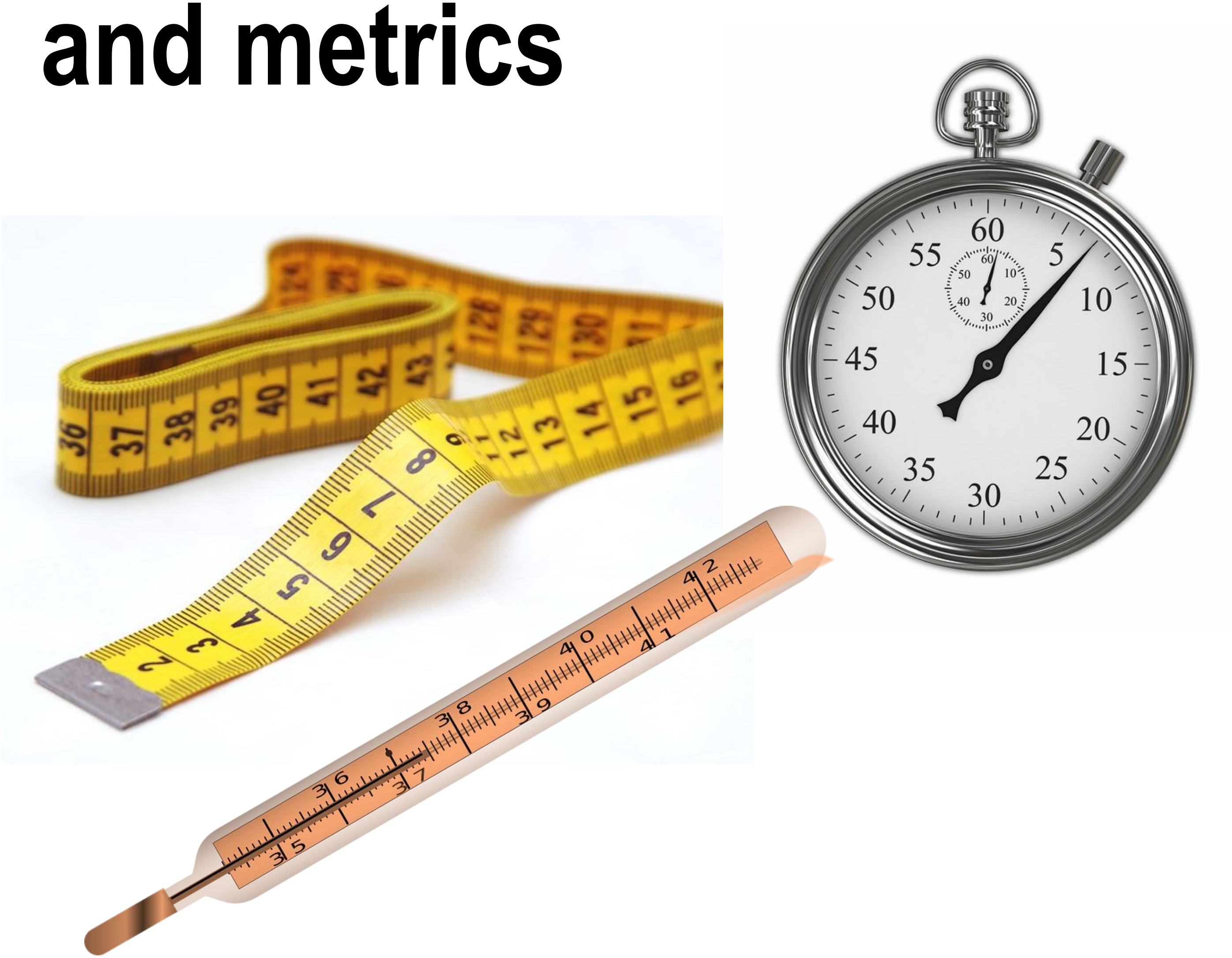
Software Quality Control

- ❑ Providing the proof that quality has been achieved
- ❑ It aims to identify and fix defects
- ❑ It is a validation process
- ❑ It always involves executing a program
- ❑ It is a Reactive and a Corrective initiative
- ❑ It is the procedure that verifies the deliverables
- ❑ It is predominantly in the testing phase
- ❑ It plays the role of the ‘second opinion’
- ❑ Its main motive is to identify defects or bugs in the system.
- ❑ It requires the involvement of the Testing team
- ❑ Executed using Statistical Quality Control methods

Metrics for Software Processes

- ❑ Why metrics
  - ▯ To characterize
  - ▯ To evaluate
  - ▯ To predict
  - ▯ To improve

Role of measurement and metrics



Measurement : It's an indication of the size, quantity, amount or dimension.

- □ Cost
- □ Defect
- □ Function point
- □ Hour
- □ Kilometre
- □ Lines of code
- □ Meter
- □ Seconds
- □ Story point

Role of measurement and metrics

Speeding Ticket

Metric : It is a measurement of degree or measure for something.

A metric is standard of measure of a degree.

□ Cost/story point
□ Defect density – defects / LOC
□ Defect density – defects / story point
□ Function point / person hour
□ Lines of code/man days
□ % Comment line

□ Is used in decision making
□ Is used in rating/ranking

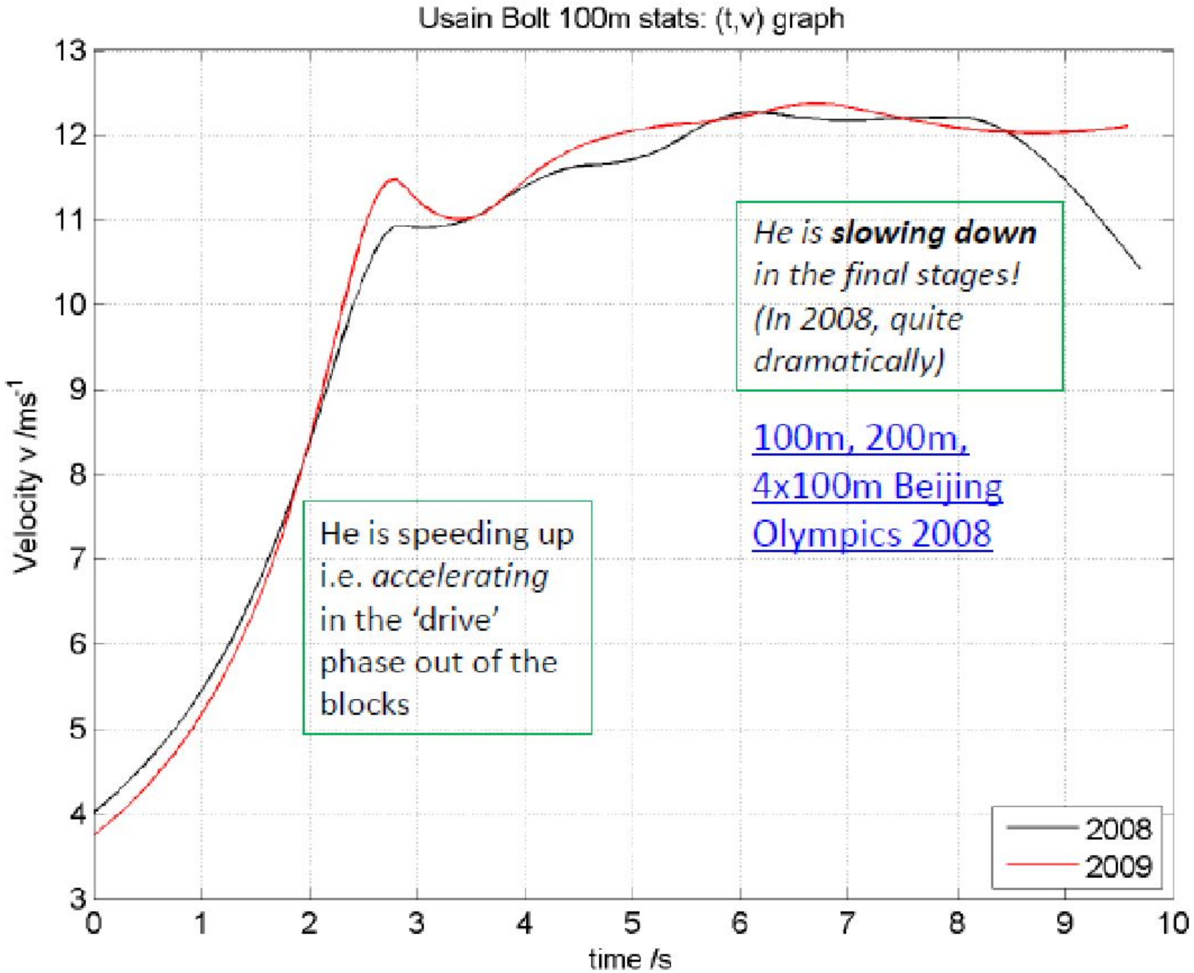
Agile metrics

https://en.wikipedia.org/wiki/Athletics_in_Jamaica

http://www.eclecticon.info/index_htm_files/Kinematics%20of%20Usain%20Bolt.pdf

- □ Agile Velocity
- □ Burndown Rate
- □ Code Coverage
- □ Cycle Time
- □ Lead Time
- □ Deployment Flags
- □ Unseen Defects

Story points/sprint



*He is **slowing down** in the final stages! (In 2008, quite dramatically)*

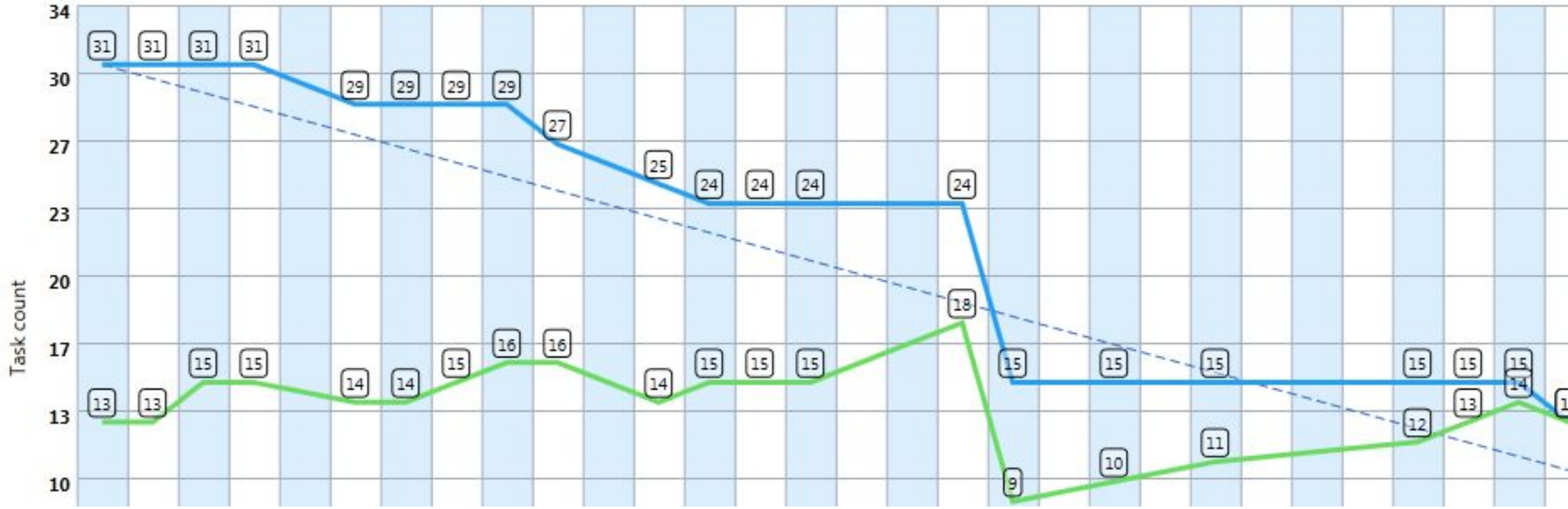
100m, 200m, 4x100m Beijing Olympics 2008

He is speeding up i.e. *accelerating* in the 'drive' phase out of the blocks

Agile metrics

- □ Agile Velocity
- □ **Burndown Rate**
- □ Code Coverage
- □ Cycle Time
- □ Lead Time
- □ Deployment Flags
- □ Unseen Defects

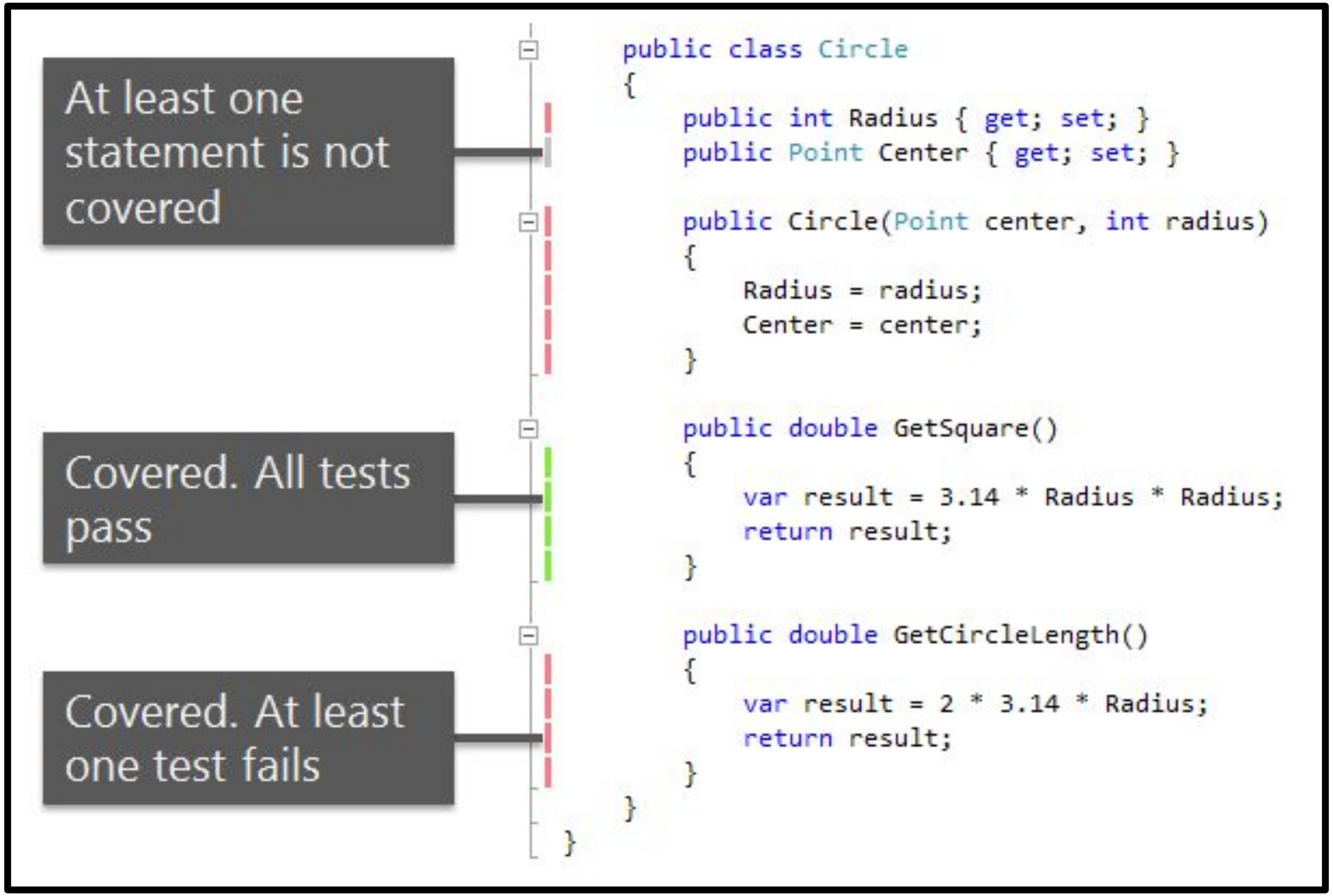
Remaining Story Points/ tasks / effort in hrs



https://commons.wikimedia.org/wiki/File:Burndown_chart.png

Agile metrics

- ❑ Agile Velocity
- ❑ Burndown Rate
- ❑ **Code Coverage**
- ❑ Cycle Time
- ❑ Lead Time
- ❑ Deployment Flags
- ❑ Unseen Defects



At least one statement is not covered

Covered. All tests pass

Covered. At least one test fails

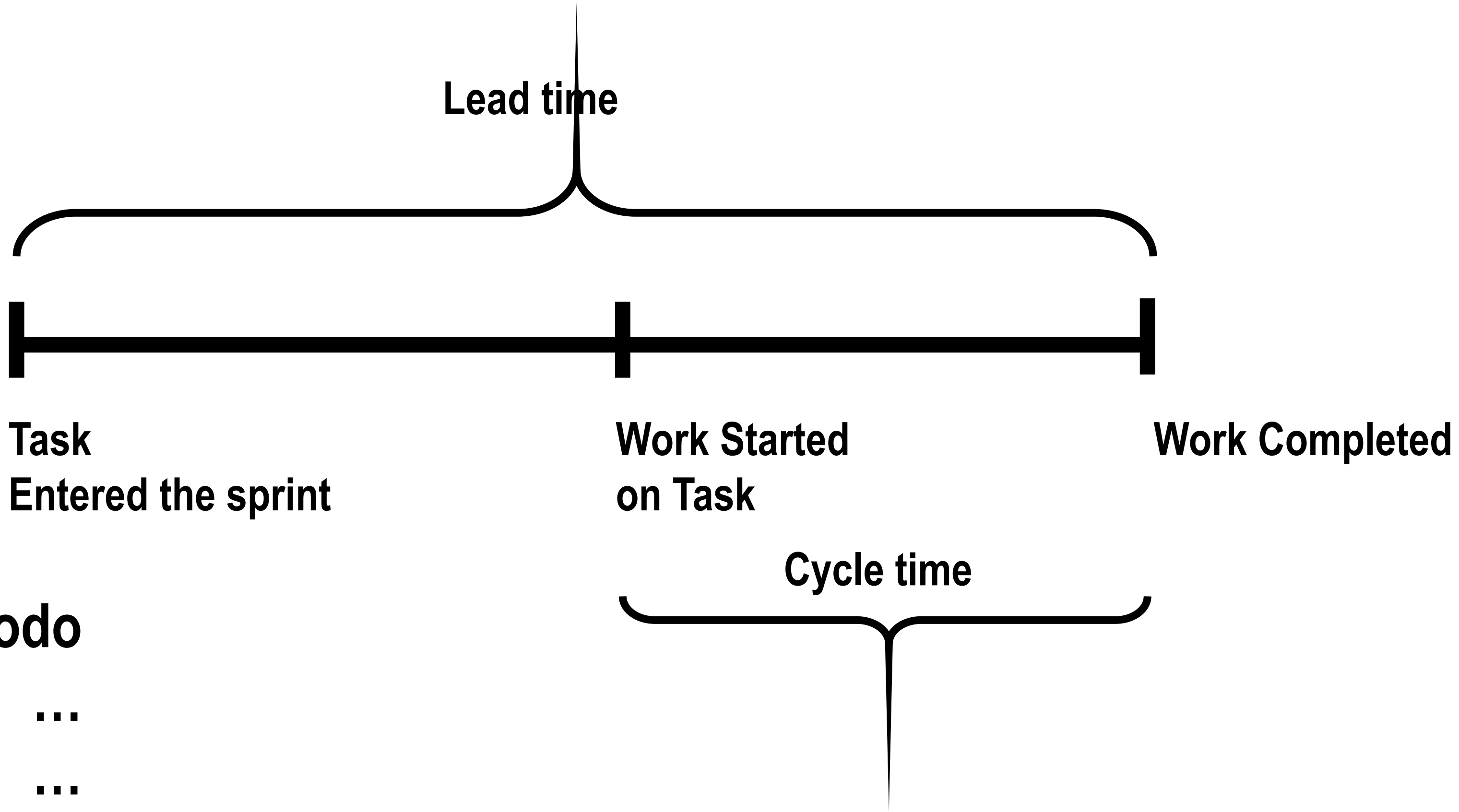
```
public class Circle
{
    public int Radius { get; set; }
    public Point Center { get; set; }

    public Circle(Point center, int radius)
    {
        Radius = radius;
        Center = center;
    }

    public double GetSquare()
    {
        var result = 3.14 * Radius * Radius;
        return result;
    }

    public double GetCircleLength()
    {
        var result = 2 * 3.14 * Radius;
        return result;
    }
}
}
```

% Lines of Code

https://blog.jetbrains.com/dotnet/2017/01/18/new-code-coverage-highlighting-in-dotcover-2016-3/

Agile metrics

- ❑ Agile Velocity
- ❑ Burndown Rate
- ❑ Code Coverage
- ❑ **Cycle Time**
- ❑ **Lead Time**
- ❑ Deployment Flags
- ❑ Unseen Defects



Lead time

Task
Entered the sprint

Work Started
on Task

Work Completed

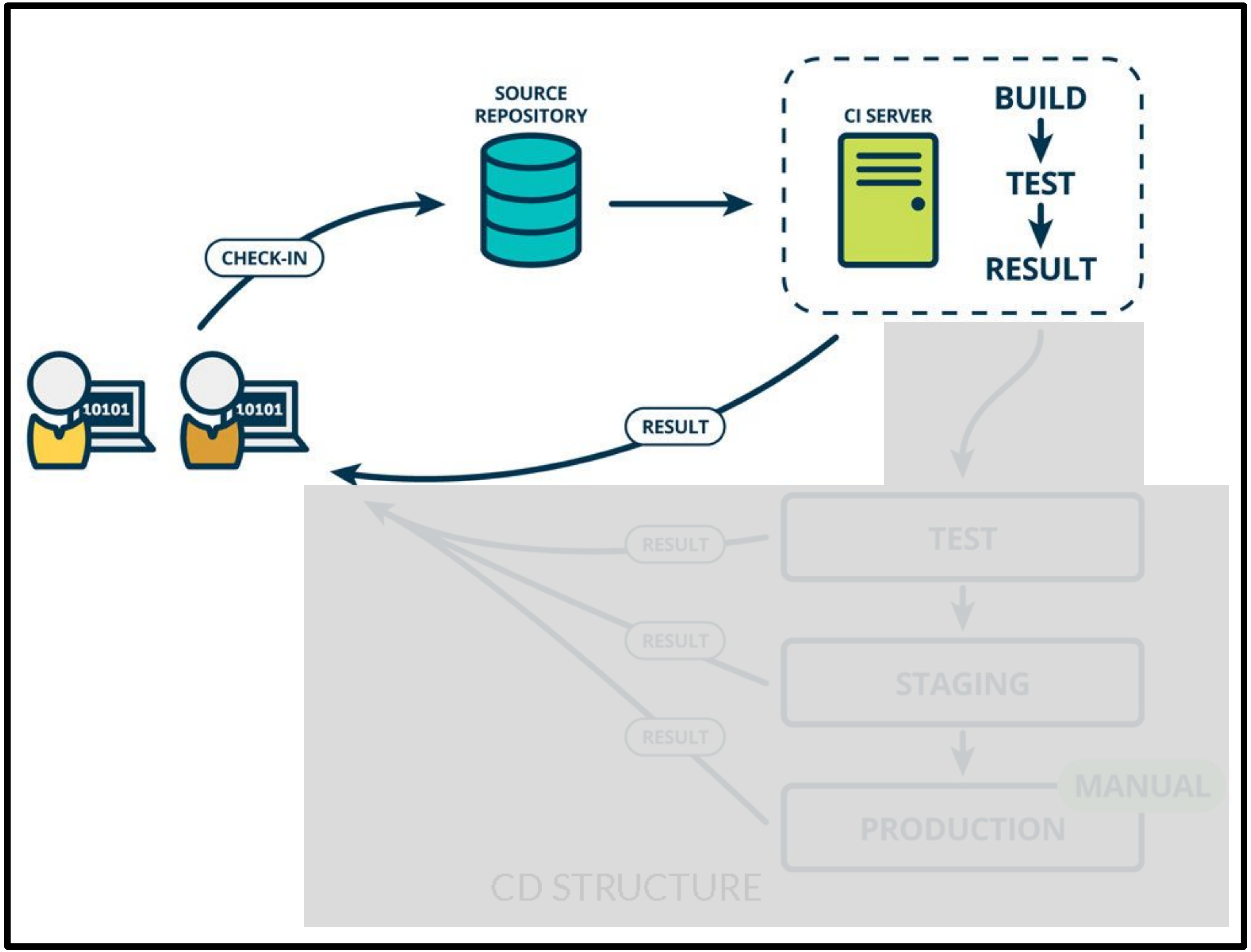
Cycle time

Todo

- ❑ …
- ❑ …
- ❑ …

Agile metrics

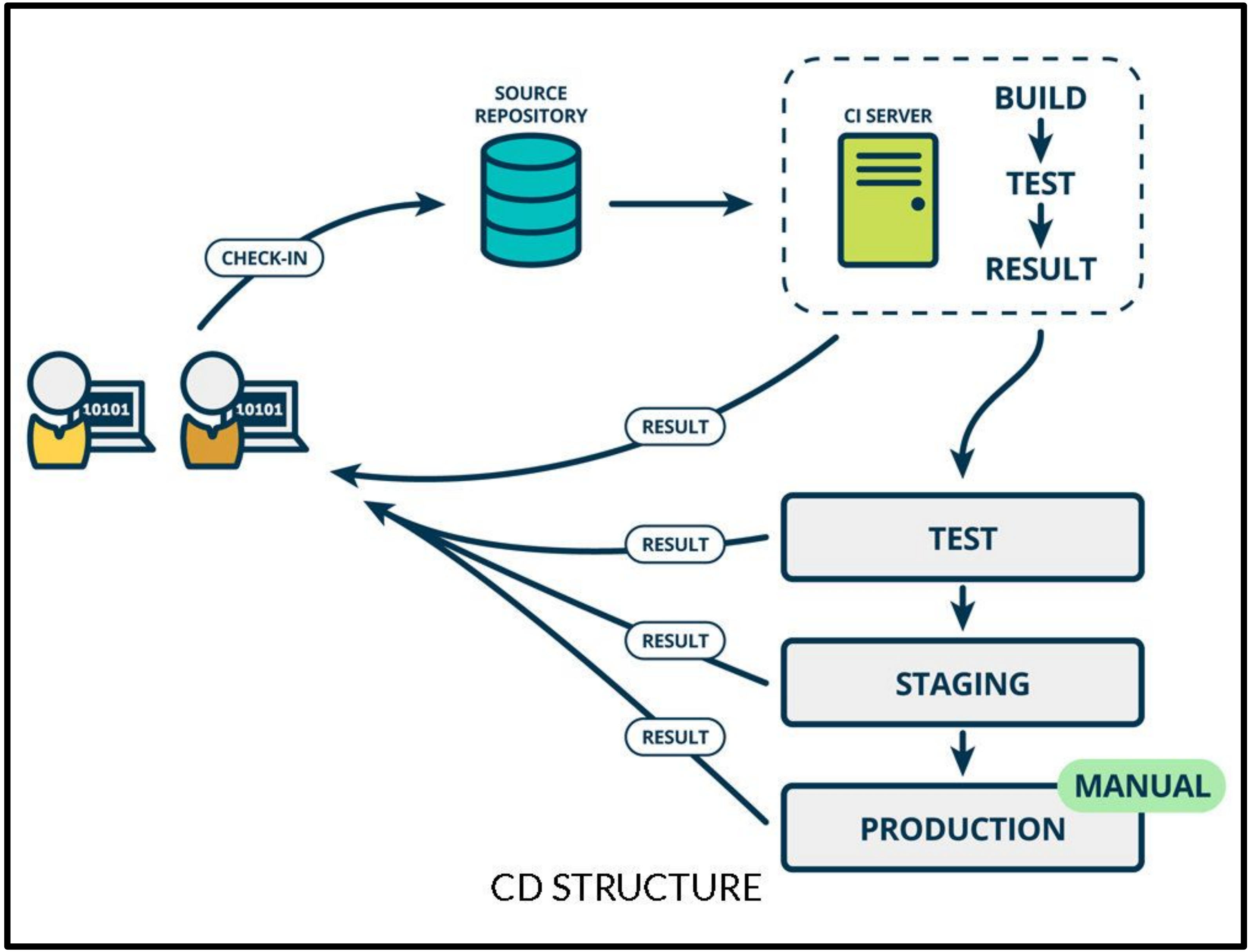
- ❑ Agile Velocity
- ❑ Burndown Rate
- ❑ Code Coverage
- ❑ Cycle Time
- ❑ Lead Time
- ❑ **Deployment Flags**
- ❑ Unseen Defects



https://www.yudiz.com/welcome-devops-prevent-defects/

Agile metrics

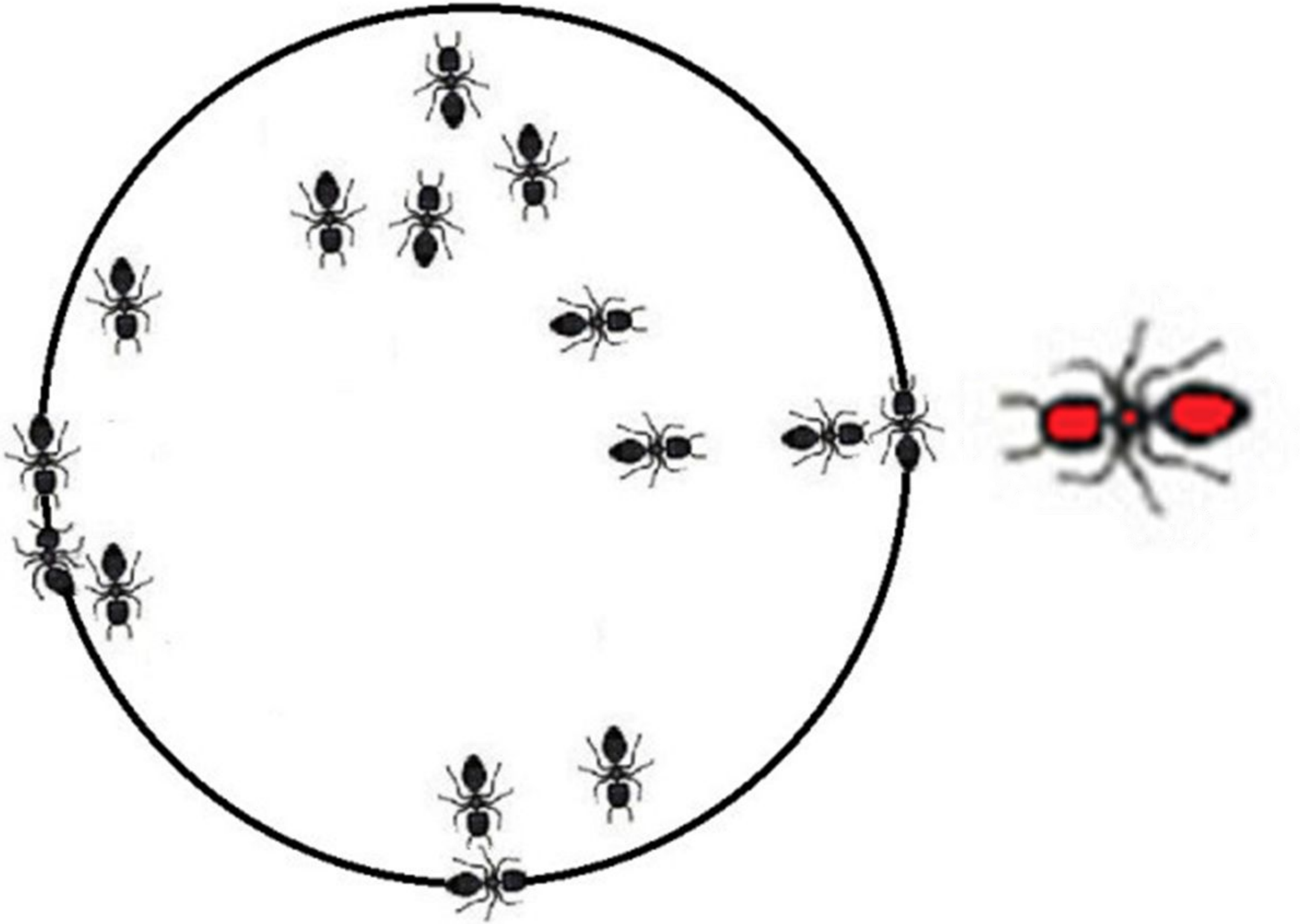
- ❑ Agile Velocity
- ❑ Burndown Rate
- ❑ Code Coverage
- ❑ Cycle Time
- ❑ Lead Time
- ❑ **Deployment Flags**
- ❑ Unseen Defects



https://www.yudiz.com/welcome-devops-prevent-defects/

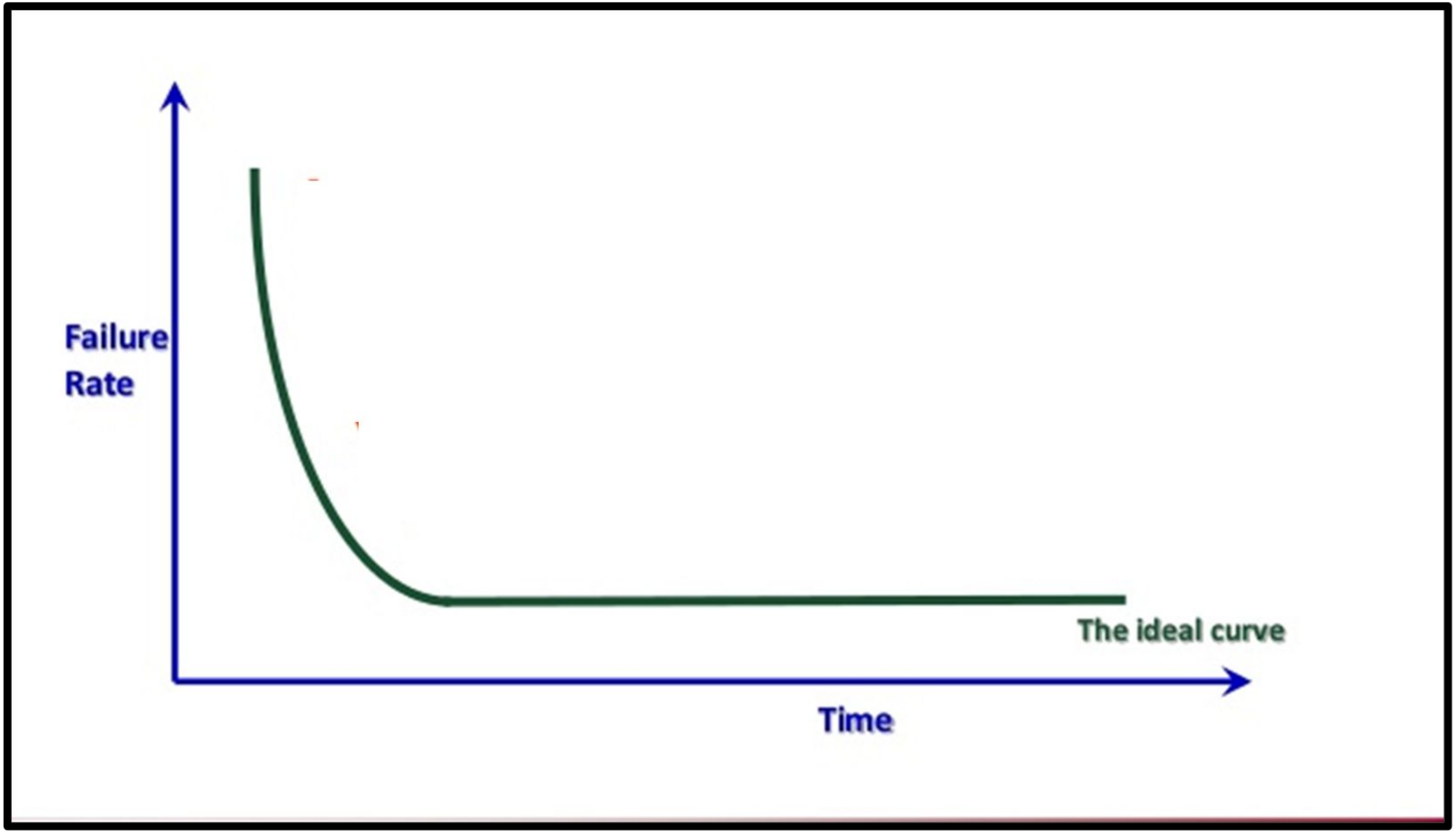
Agile metrics

- ❑ Agile Velocity
- ❑ Burndown Rate
- ❑ Code Coverage
- ❑ Cycle Time
- ❑ Lead Time
- ❑ Deployment Flags
- ❑ **Unseen Defects**



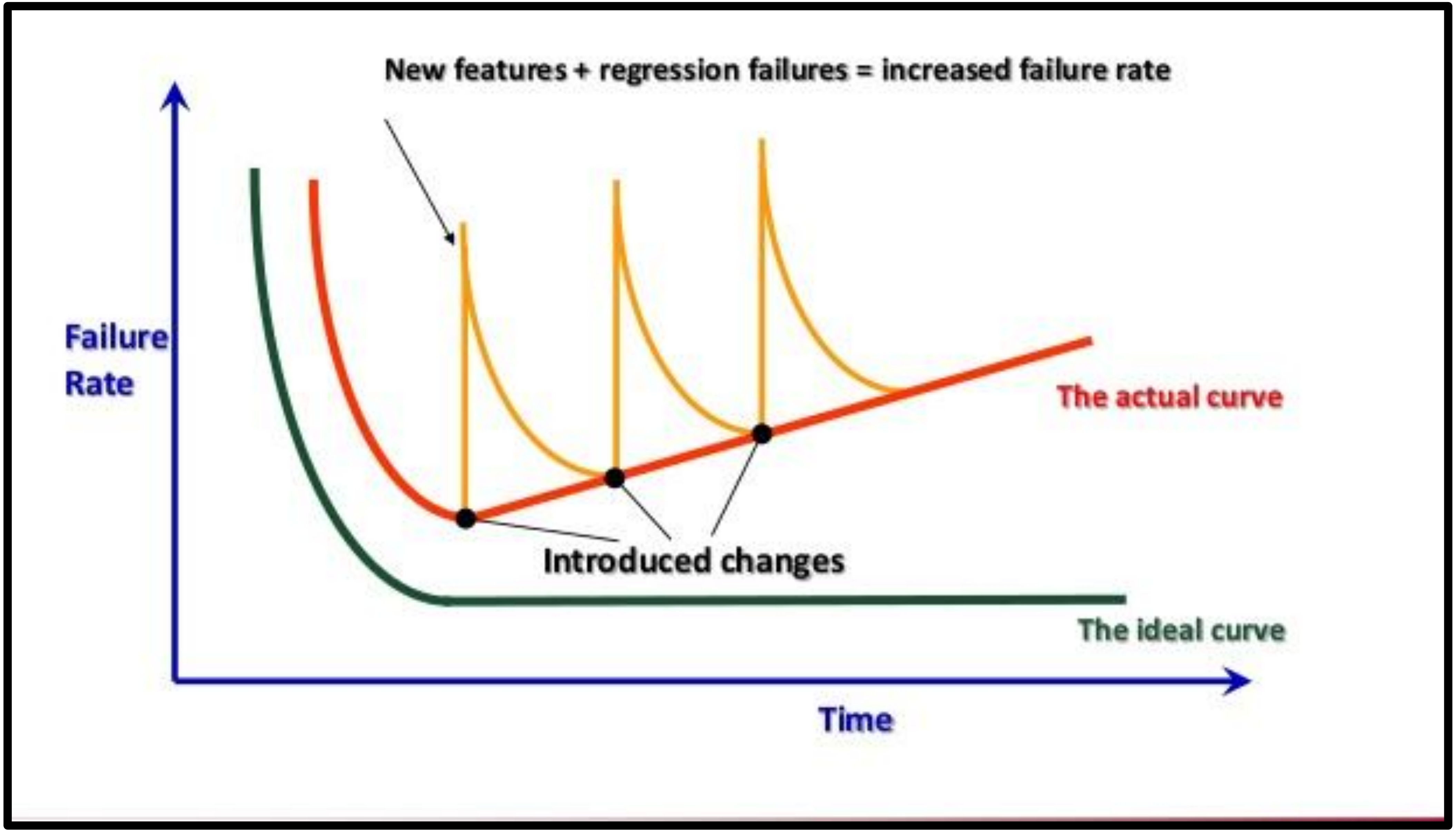
Metrics for Software Reliability

https://www.slideshare.net/DrAmirTomer/swise-arc2015



Metrics for Software Reliability

https://www.slideshare.net/DrAmirTomer/swise-arc2015

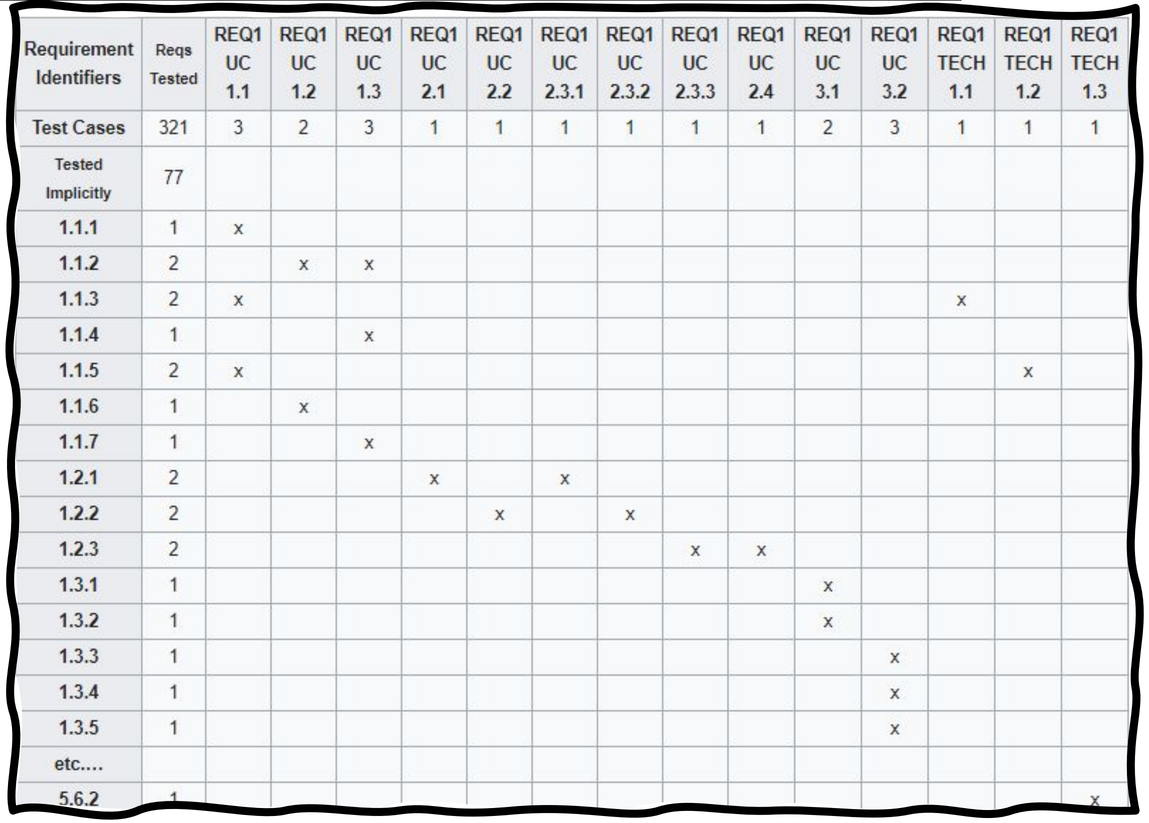


Metrics for Software Reliability

- ❑ Product metrics
- ❑ Project management metrics
- ❑ Process metrics
- ❑ Fault and failure metrics

Metrics for Software Reliability

- ❑ Product metrics
  - ❑ Requirements Traceability Matrix
  - ❑ Design documents
  - ❑ Algorithms/Flowchart
- ❑ Project management metrics
- ❑ Process metrics
- ❑ Fault and failure metrics



Requirement Identifiers	Reqs Tested	REQ1 UC 1.1	REQ1 UC 1.2	REQ1 UC 1.3	REQ1 UC 2.1	REQ1 UC 2.2	REQ1 UC 2.3.1	REQ1 UC 2.3.2	REQ1 UC 2.3.3	REQ1 UC 2.4	REQ1 UC 3.1	REQ1 UC 3.2	REQ1 TECH 1.1	REQ1 TECH 1.2	REQ1 TECH 1.3
Test Cases	321	3	2	3	1	1	1	1	1	1	2	3	1	1	1
Tested Implicitly	77														
1.1.1	1	x													
1.1.2	2		x	x											
1.1.3	2	x											x		
1.1.4	1			x											
1.1.5	2	x												x	
1.1.6	1		x												
1.1.7	1			x											
1.2.1	2				x		x								
1.2.2	2					x		x							
1.2.3	2								x	x					
1.3.1	1										x				
1.3.2	1										x				
1.3.3	1											x			
1.3.4	1											x			
1.3.5	1											x			
etc....															
5.6.2	1														x

Metrics for Software Reliability

- □ Product metrics
- □ Project management metrics
- □ Process metrics
- □ Fault and failure metrics

- □ Change Management
- □ Cost control
- □ Risk evaluation
- □ Communication

Metrics for Software Reliability

- □ Product metrics
- □ Project management metrics
- □ Process metrics
  - □ Audit Checklist
  - □ Code completion
  - □ Check in / Check out
  - □ Dev-ops Scripts
- □ Fault and failure metrics

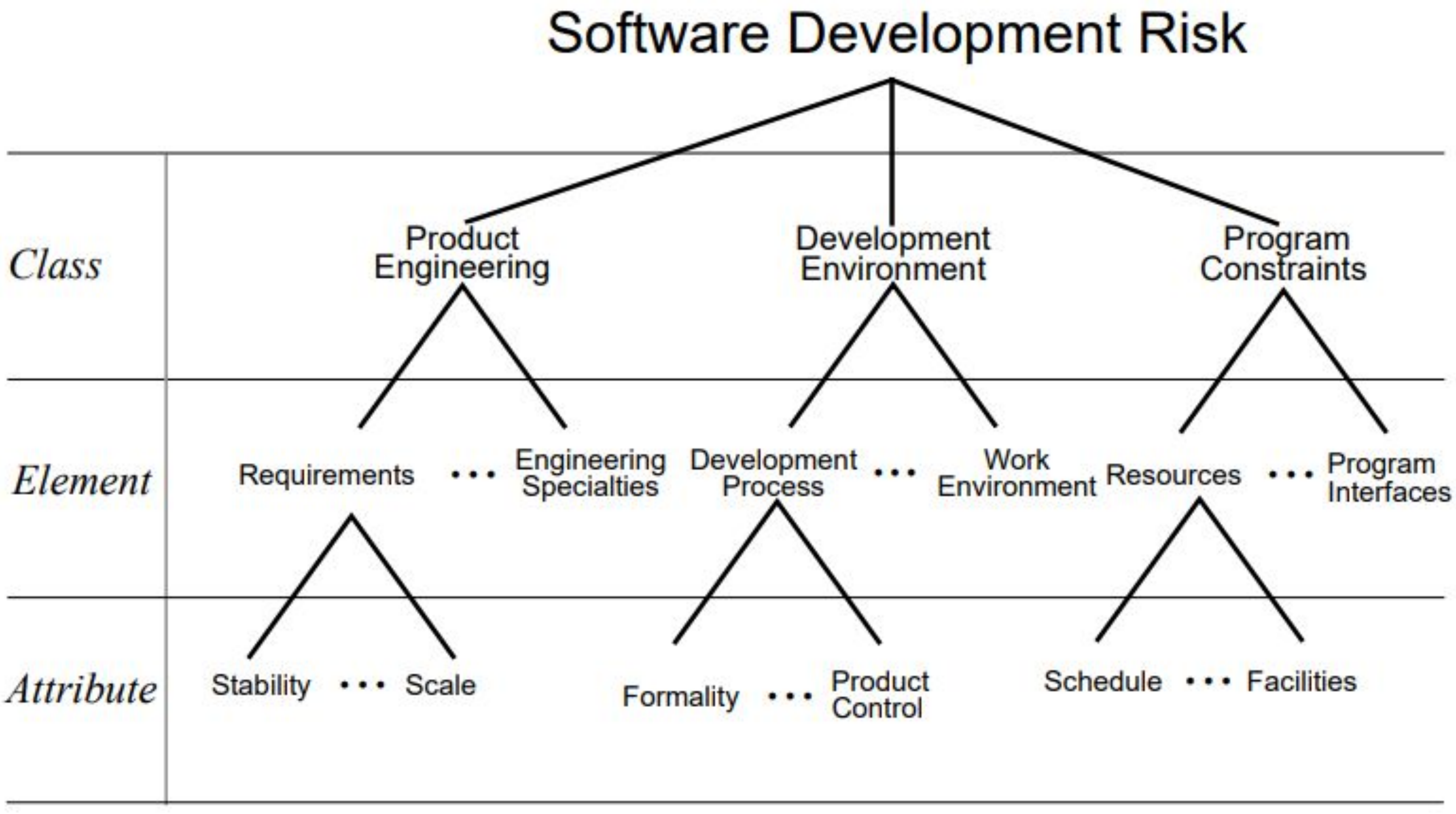
Metrics for Software Reliability

- Product metrics
- Project management metrics
- Process metrics
- Fault and failure metrics
    - Defects
    - Ambiguities
    - Oversights or misinterpretation of the specification
    - Carelessness or incompetence in writing code
    - Inadequate testing
    - Incorrect or unexpected usage of the software
    - Other unforeseen problems

Metrics for Software Reliability

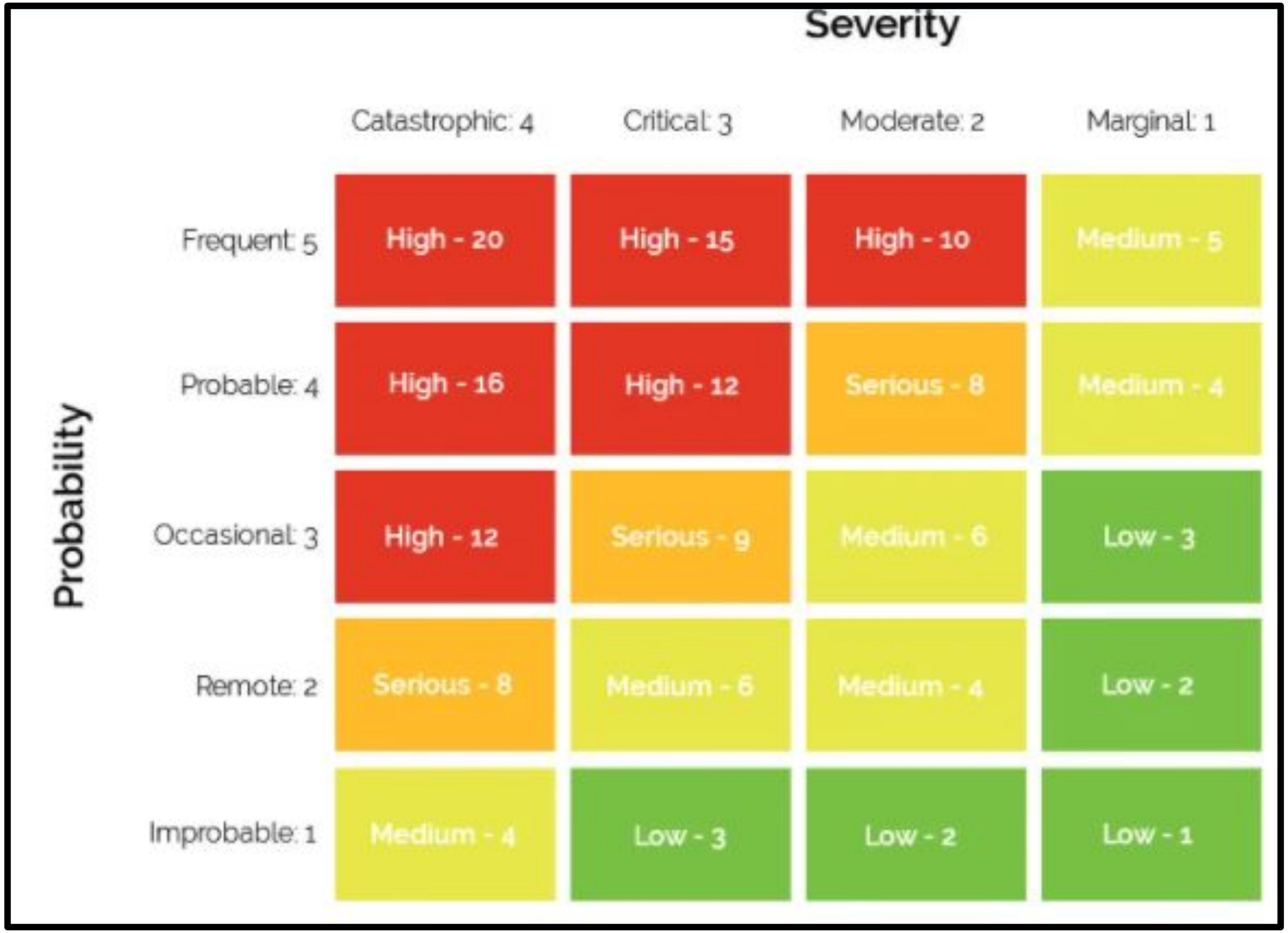
- Determine when the software is approaching failure-free execution
  - Mean time between failures - MTBF
  - Mean time to repair - MTTR
- Analyze errors and defects
- Predict using MTBF

Risk Assessment Framework



https://resources.sei.cmu.edu/asset_files/TechnicalReport/1993_005_001_16166.pdf

Risk



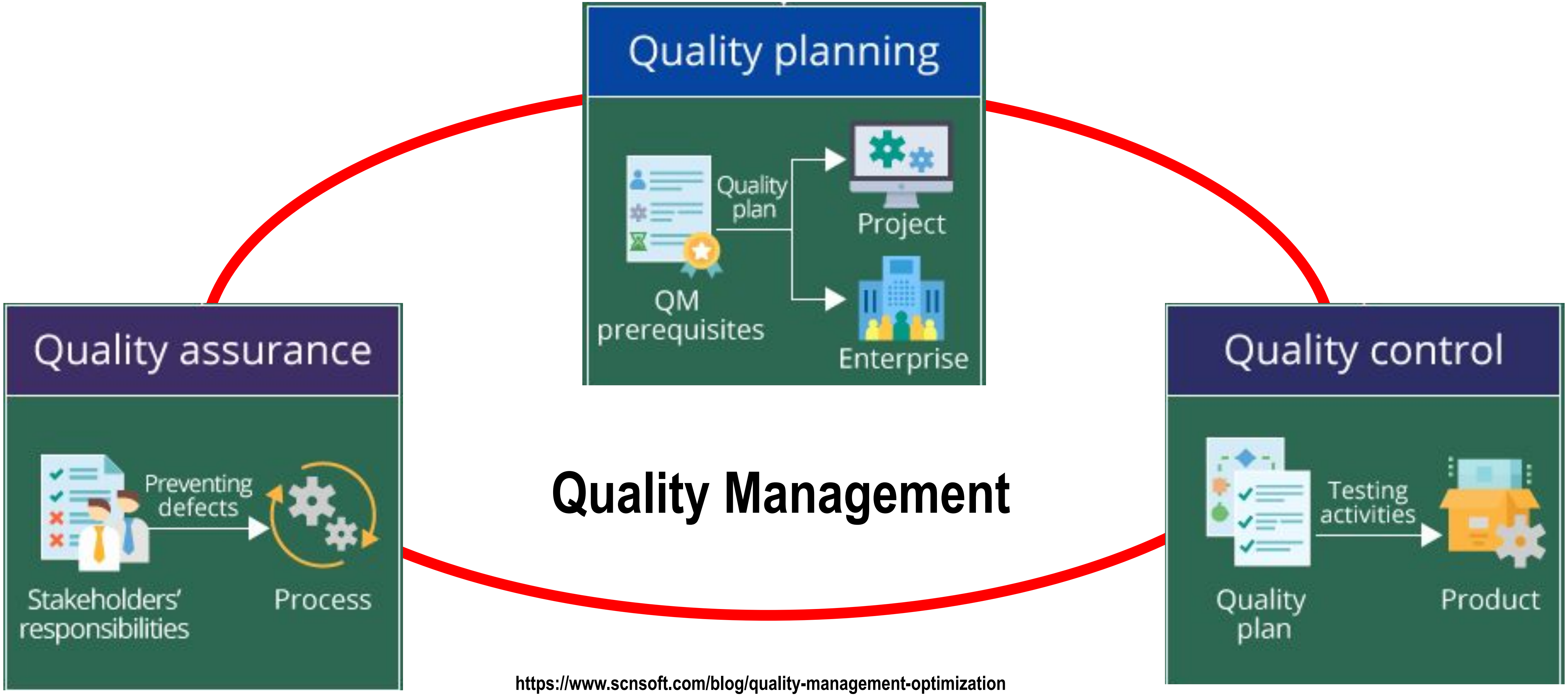
Severity

Probability

	Catastrophic: 4	Critical: 3	Moderate: 2	Marginal: 1
Frequent: 5	High - 20	High - 15	High - 10	Medium - 5
Probable: 4	High - 16	High - 12	Serious - 8	Medium - 4
Occasional: 3	High - 12	Serious - 9	Medium - 6	Low - 3
Remote: 2	Serious - 8	Medium - 6	Medium - 4	Low - 2
Improbable: 1	Medium - 4	Low - 3	Low - 2	Low - 1

https://www.industrysafe.com/blog/risk-matrix-calculations-severity-probabili
y-and-risk-assessment/

Software Quality Management



https://www.scnsoft.com/blog/quality-management-optimization

We covered the following

❑ Understand the basics of Software Quality

❑ Differentiate Software Quality Control and Software Quality Assurance

❑ Understand the use of metrics for Software Quality

❑ Understand Software Quality Control Processes

❑ Integrating metrics within software quality processes

❑ Understand software reliability and metrics