An End-to-End Quality Management Framework to improve Software Quality and Productivity

Vincent Yip
Senior Development Manager
Thomson Reuters Hong Kong Ltd

Agenda

- Background
- Initiatives
- End-to-end Quality Management Framework
  - Predict Defects
  - Identify Contributing Factors
  - Plan Peer Reviews
  - Manage defects found against target
- Results

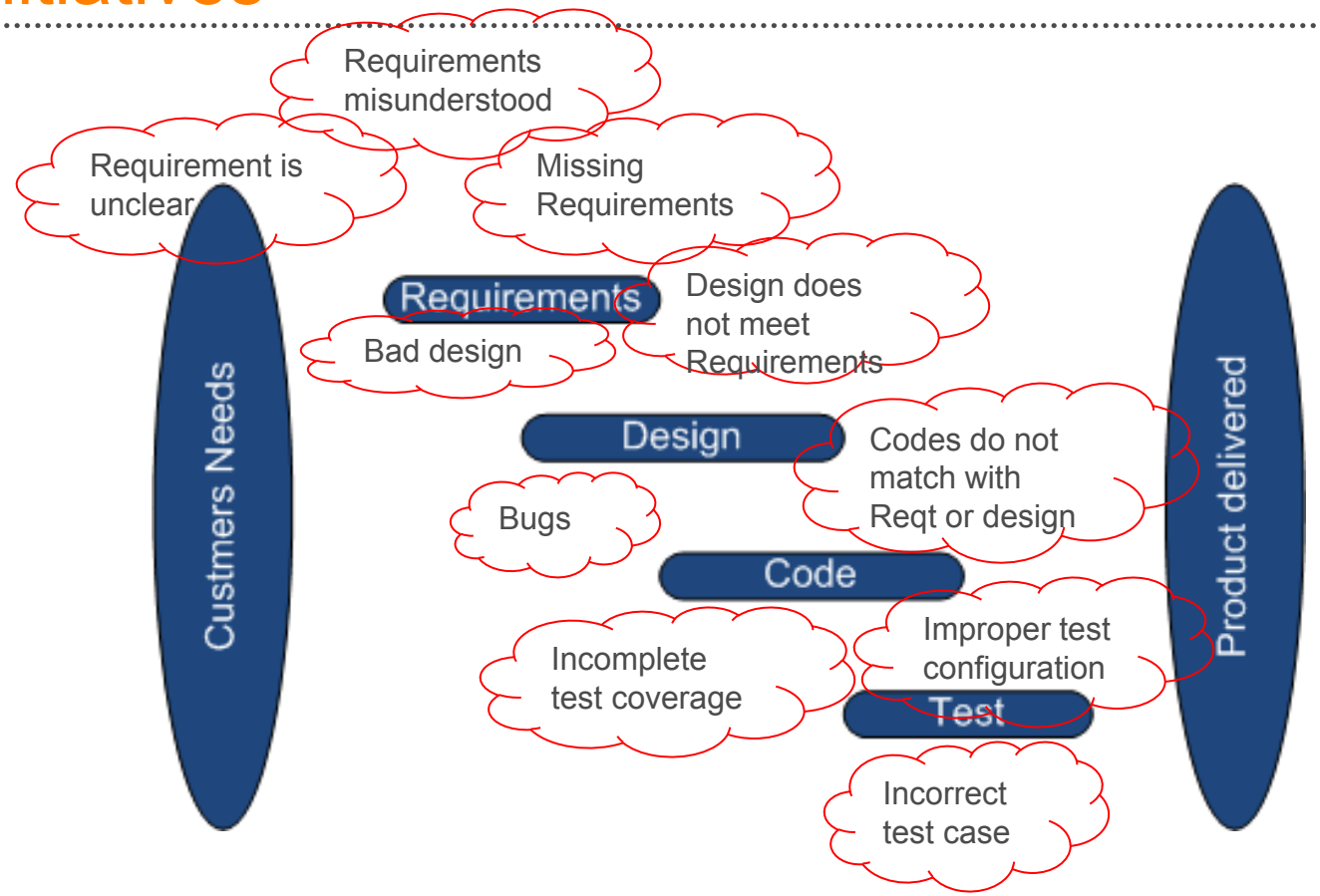
Background - 1

- One of the major development centers in Thomson Reuters
- Develop a number of key systems that deliver real-time financial data and news to clients around the world for
  - Low latency, high frequency algo trading
  - Human desktop consumption.
- Develop a framework for remote monitoring and management of thousands of servers in the real-time data network.

Background - 2

- Data accuracy and system reliability are critical to our business
- Software quality is the key success factor of our development group
- Adopted CMMI as our software quality and process improvement framework since 1996
- Used Six Sigma mechanisms and tools to analyse and improve processes

Initiatives



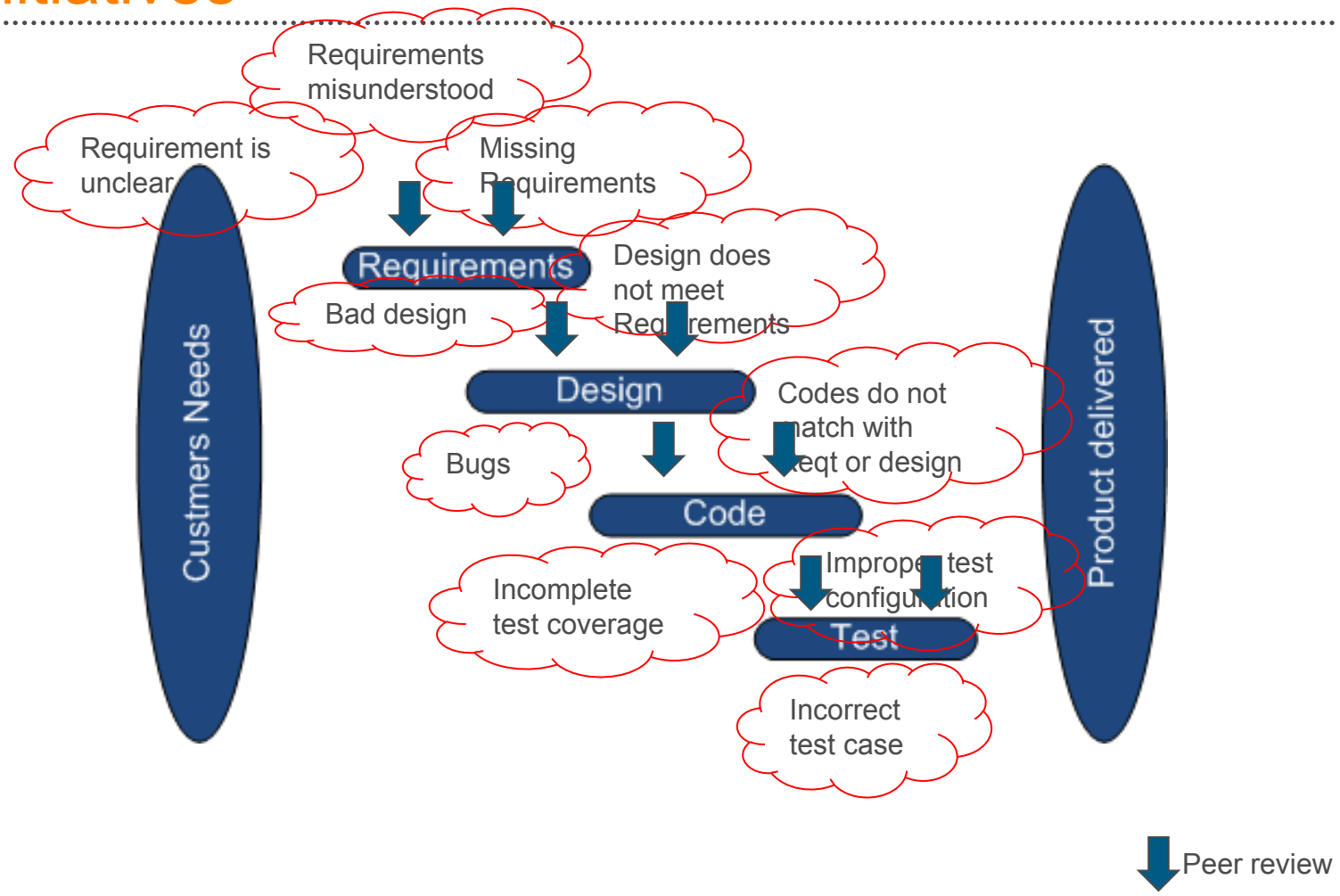
Initiatives

- If we rely on testing only to uncover defects
  - Uncertainty on the quality of the software product
  - Lots of defects may be uncovered in later phases of the project
  - Lots of rework late in the project
  - Fire fighting … more pressure to deliver the product
  - Project may be delayed or released with deficiency

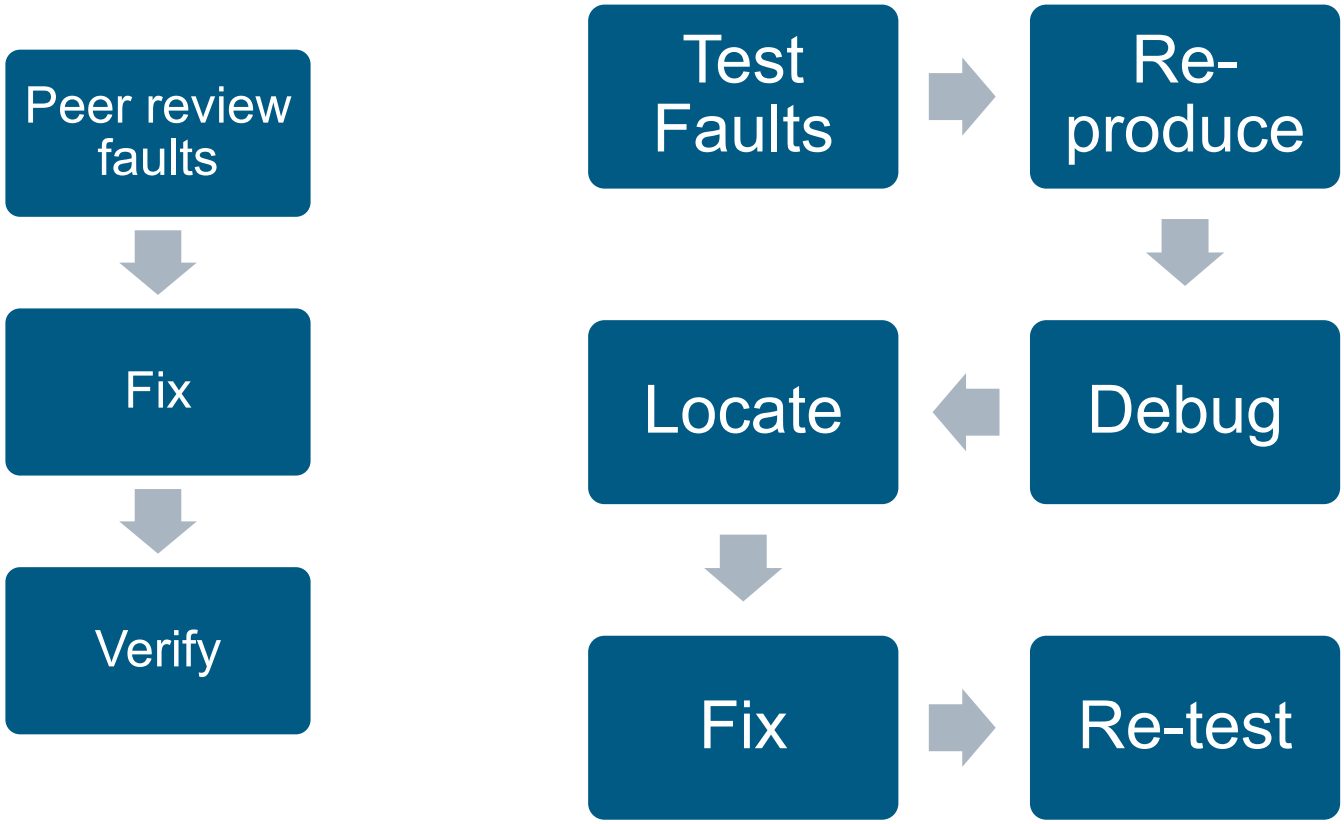
Initiatives

- Adopt peer review as a light weight process to verify work products
  - Quality gates before moving to the next phase
  - Uncover defects earlier
    - contain defect from where it is originated
    - reduce rework effort
  - Align and improve understanding of the work

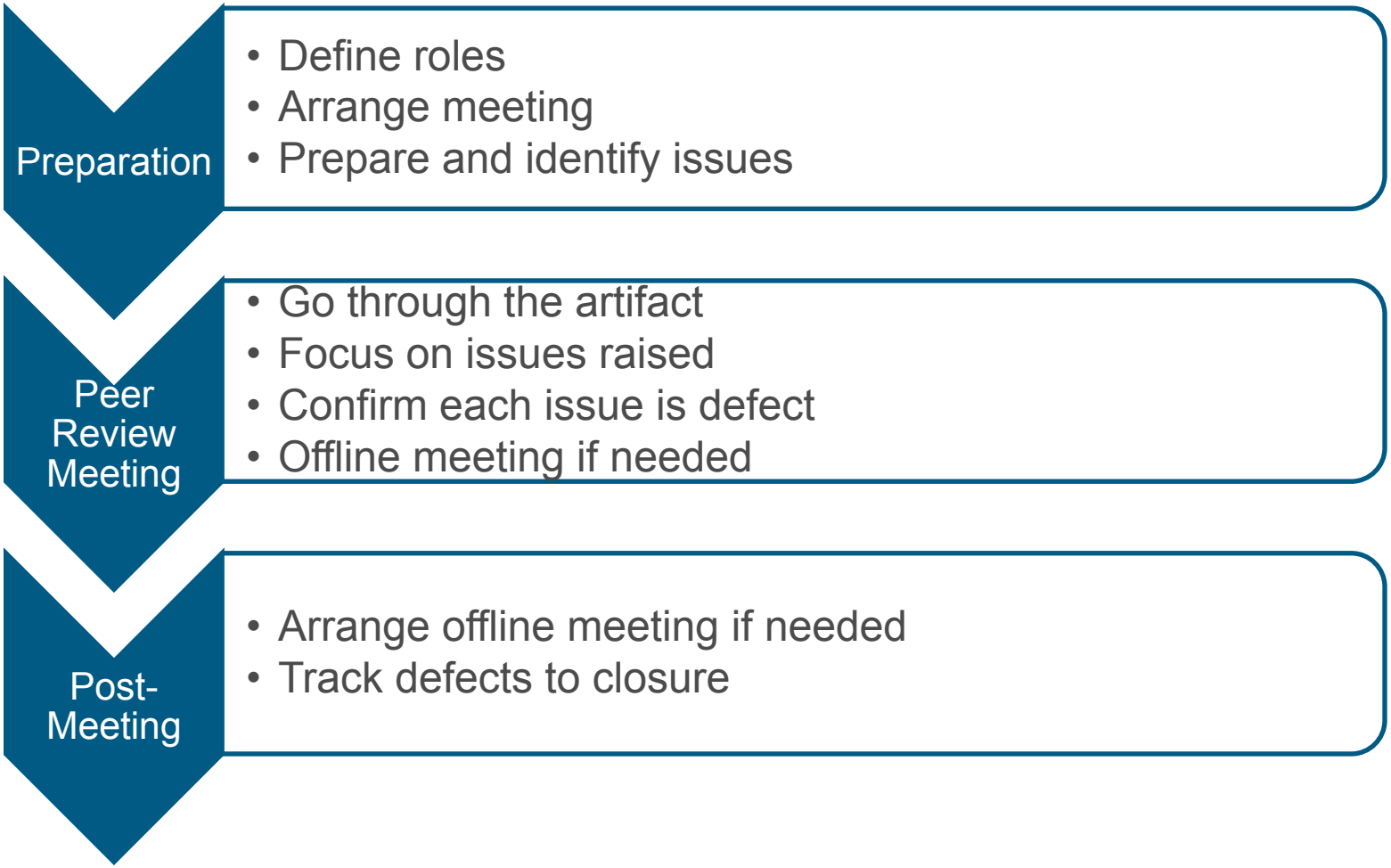
Initiatives



Peer Review vs Test

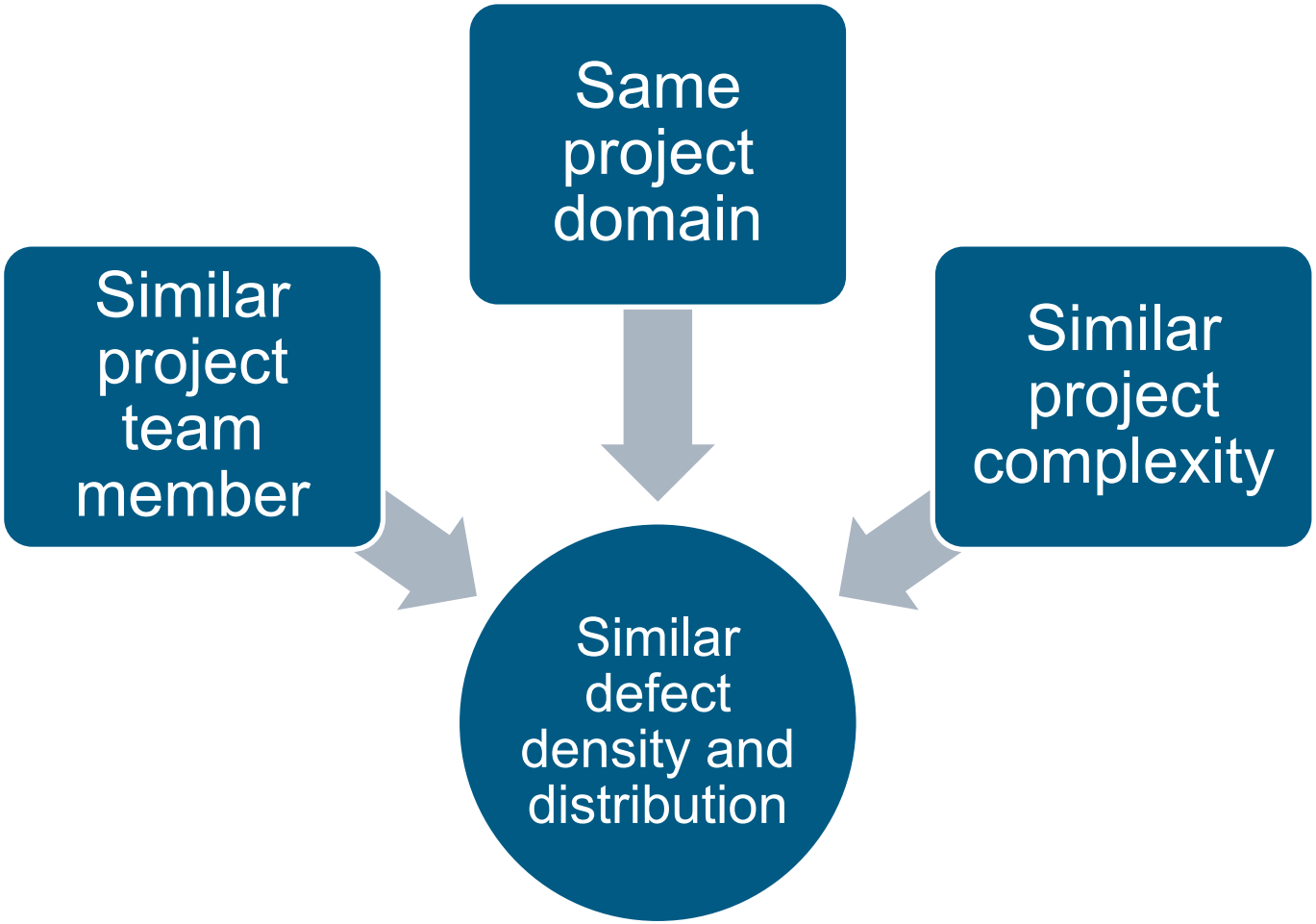


Standard Peer Review Process



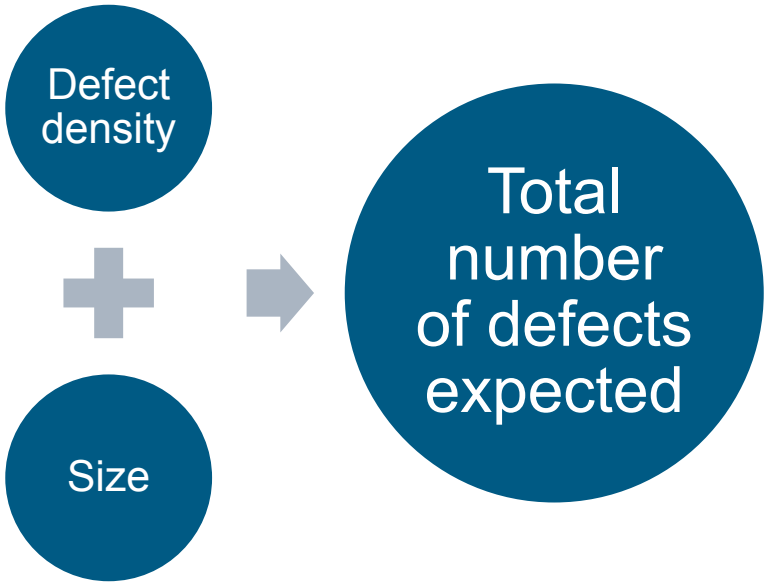
END TO END QUALITY MANAGEMENT FRAMEWORK

Predict Defects - Assumption



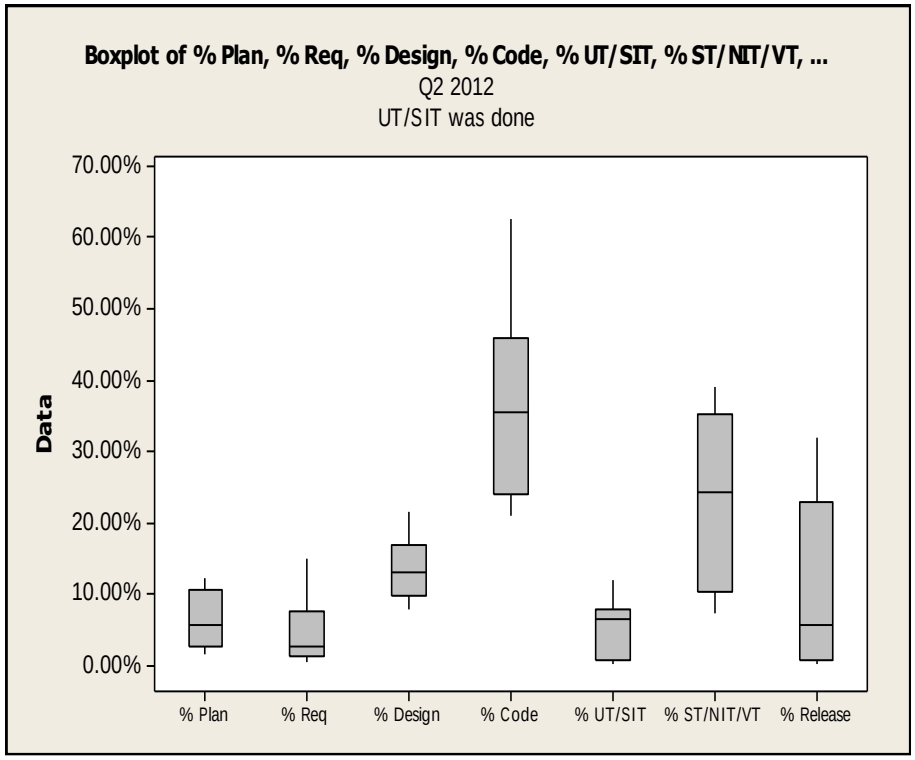
Predict Defects – Total Defects Generated

- Estimate defect density based on previous project data
- Estimate current project size
- Predict the number of defects produced throughout the development cycle



Predict Defects – Estimate the distribution

- Understand the historical % distribution of defects
- Compare the project effort distribution against the historical data



Predict Defects – Target Defects

Total Defects expected

+

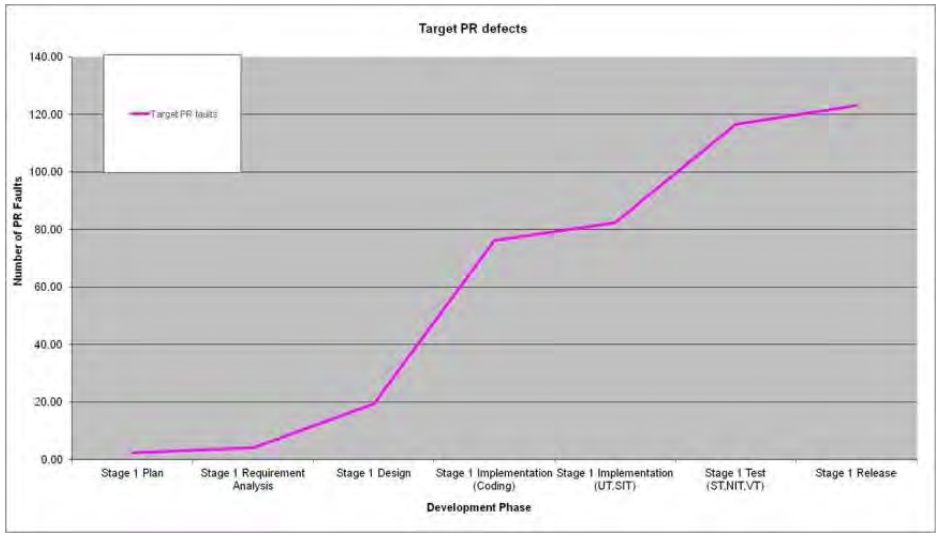
Target % defects captured in PR

+

% distributio n in each phase

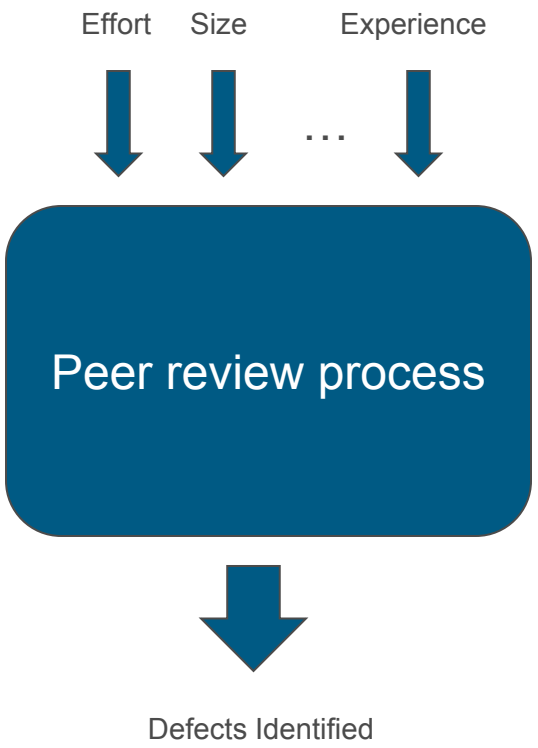
→

Number of defects expected in each phase



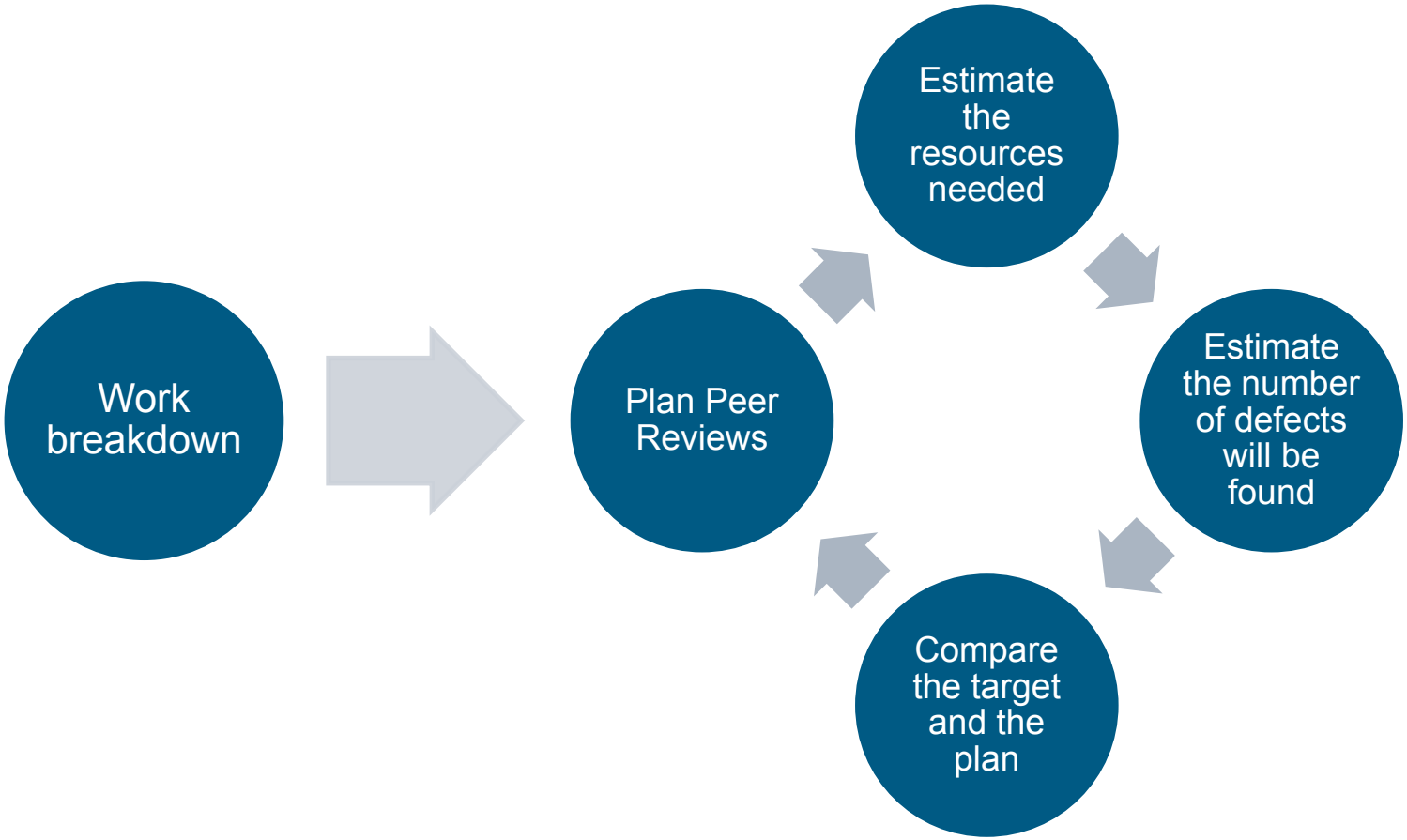
Identify Contributing Factors

- Analyse peer review using regression analysis
  - Input factors
  - Output – defects identifed



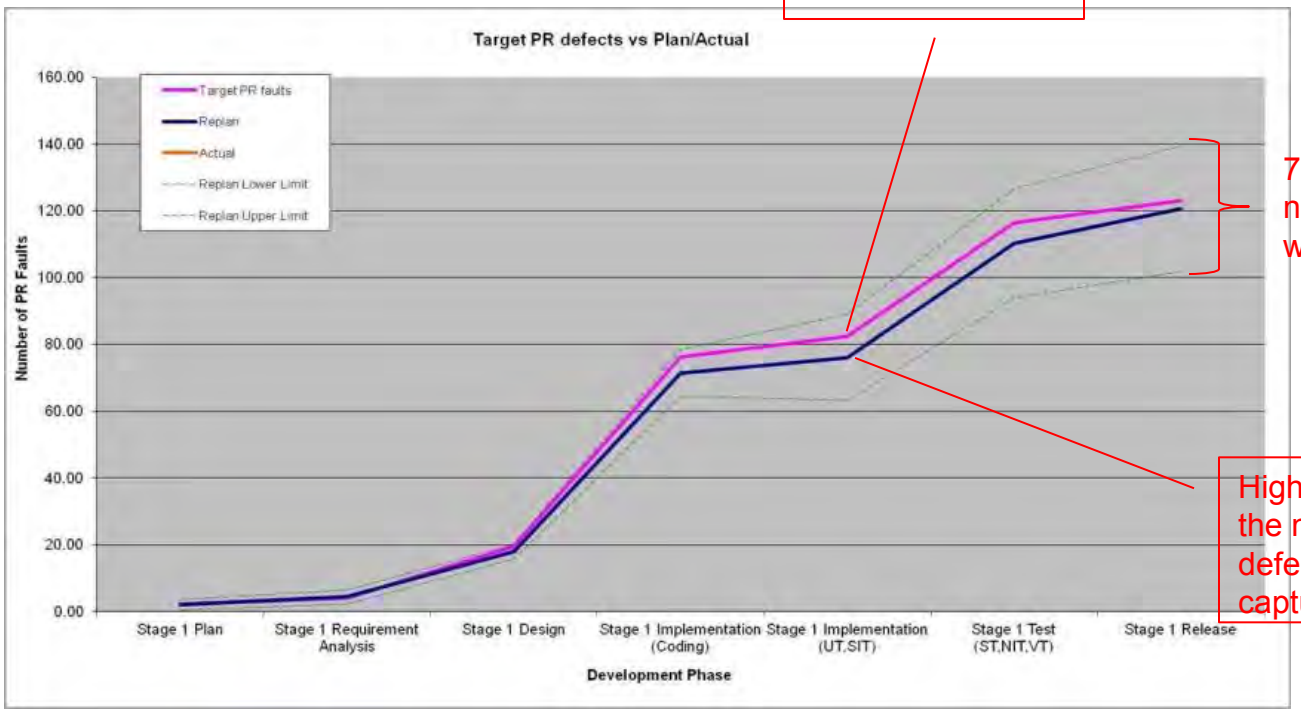
Number of defects identified = a + b * prep time + c * PR meeting time + …

Plan Peer Reviews and Defects Captured



Plan Peer Reviews and Defects Captured

- Estimate defects can be captured from the planned peer reviews
- 70% prediction interval



Number of defects expected to be generated.

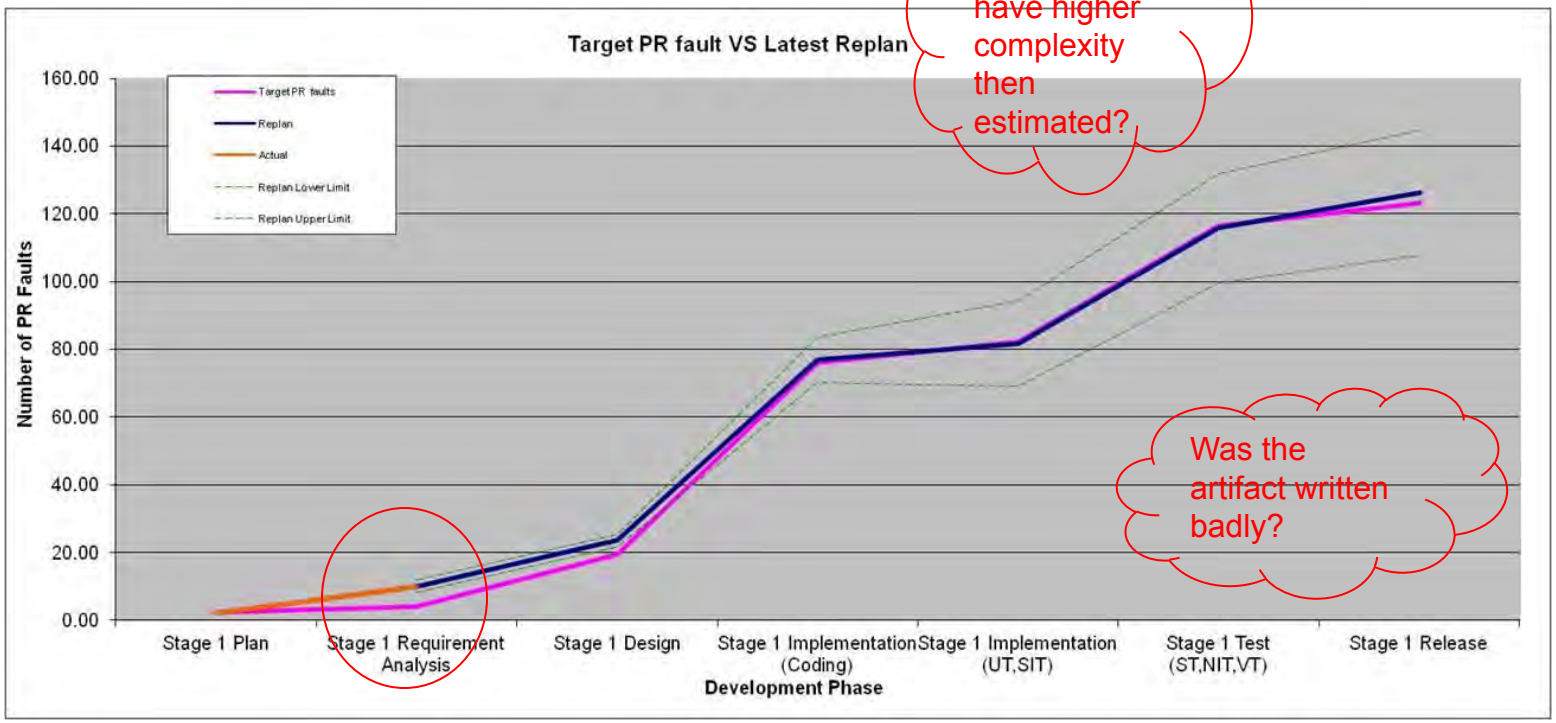
70% chance the number of defects will be captured

Highest possibility the number of defects will be captured

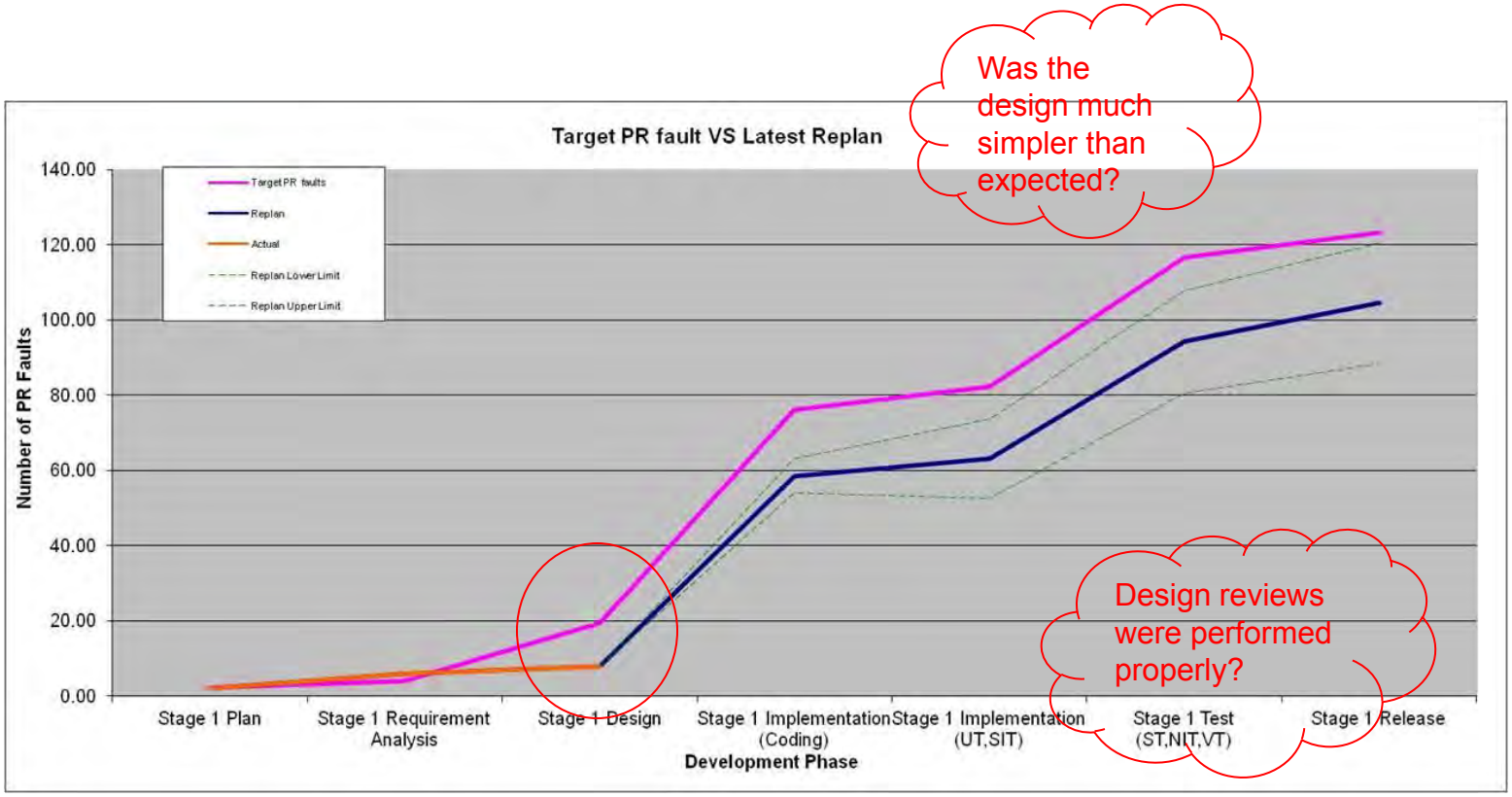
Manage Defects Found Against Target

- Track actual results against the plan
- Revisit project estimation if escaped out of the 70% prediction interval
  - Complexity?
  - Project size?
  - Distribution of work and defects?
  - Peer reviews were performed properly?

Manage Defects Found Against Target



Manage Defects Found Against Target



Questions?