
Test Management Part 2

Causal Analysis

Objective



Objective

Apply Causal
Analysis to
improve test
effectiveness

Causal Analysis



| First introduced by IBM in 1983

- IBM reports an investment of less than 1% in causal analysis achieved a 50% reduction in error rates

| Seeks to identify the root cause of defects as well as approaches to eliminate future occurrences

| Defect data is collected and analyzed

Causal Analysis Steps



Select a set of defects missed by testing to analyze

For each defect identify the probable cause it was not detected

Identify common causes among defects

Identify possible solutions to eliminate the common defect causes

Defect Cause Categories



Communication Failure

| Defect was missed due
to missing or incorrect
information

| Examples:

- Problems with requirements doc
- Last minute requirements changes not communicated to test team

Defect Cause Categories (cont'd)



Oversight

| **Defect was missed because of a failure to consider all cases or combinations when testing**

| **Examples:**

- Failing to test a feature interaction
- All combinations of inputs not addressed
- Boundary value not tested

Defect Cause Categories (cont'd)



Education

| Defect was missed because of a lack of understanding of testing methodologies or tools

| Examples:

- Design of experiments was incorrectly applied
- Tester didn't understand how automated tests were generated

Defect Cause Categories (cont'd)



Transcription

| Missed defect was a result of a simple mistake

| Examples:

- Tester failed to carefully compare expected and actual results

Determining Probable Root Causes



Five Whys

- | Ask why enough times to get to the root cause of the defect
- | Up to 5 repetitions may be needed

Example



| Consider customer reported feature interaction error

| Why was defect not detected during testing

- Feature interaction was never tested

| Why was feature interaction not tested

- Late feature addition was not communicated to test team

Developing Possible Solutions



Oversight Problems

- | Use of checklists
- | Tools to automate checking
- | Work-product templates
- | Reviews

Developing Possible Solutions (cont'd)



Education Problems

- | Just in time training
- | Tutorials
- | Proper staffing

Developing Possible Solutions (cont'd)



Communication Problems

- | **Liaisons to other groups**
- | **Change tracking system improvements**
- | **Improved documentation**
- | **Changing processes**

Developing Possible Solutions (cont'd)



Transcription Problems

| Tools to automate

| Reviews

Summary



Test Management Part 2

People Management

Objective



Objective

Apply best
practices for
maximizing test
team
performance

Sources of Leverage for Improving Software Development



Make Effective Use of Your People



| One of the most significant factors of a project's success is the ability, experience and motivation of its people

| Software test leader must do everything possible to maximize individual and team effectiveness

Responsibilities of System Test Lead



| Negotiate with project management on schedule and resources

| Establish priorities for test team

| Manage team

| Negotiate with development team on entry criteria

Basic Management Skills



| Leadership

| Communication

| Delegation

| Negotiation

| Motivation

| Problem solving

Motivation



What factors motivate software testers?

Motivation Guidelines for Leaders



| **Learn what your team's needs, goals and motives are**

| **Provide a work environment and task structure which fulfills people's needs**

| **Implement a fair reward system**

- Recognition, technical advancement, increased technical knowledge and responsibility are important factors as well as salary

| **If goals, plans, requirements and expectations are clearly communicated and understood by the team, the team members will be motivated to perform**

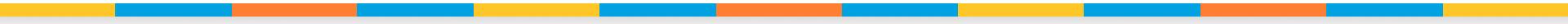
Key Motivators for Software Testers



| Pride in work

- Testers who take pride in their work have increased job satisfaction and higher quality
- Test lead must ensure that testers are “proud” of their work
- Test lead must ensure testers can accomplish their tasks

Key Motivators for Software Testers



| Pride in accomplishment

- Early success increases motivation
- Project activities should be structured to create early “accomplishments”

| Pride in contribution

- There is a strong human desire to contribute
- Recognition of the importance of each tester's contribution is critical

Pair Testing



- | Involves 2 testers exploring system input space for tests to execute
- | Analogous to pair programming in extreme programming

- | Forces communication and classification of testing strategies
- | Advantages and disadvantages

Team Development



| **It is important to improve the knowledge and skills of all team members**

| **Strategies for team development include:**

- Training
- Job rotation
- Mentoring
- Reviews, appraisals and feedback

Importance of Teamwork



| A successful project requires skilled individuals who work effectively together as a team

| Team success is driven by the clear communication of project goals

- The team must understand what it is trying to accomplish
- Individual team members must understand how they contribute to the team's goal

Characteristics of Good Teams

(derived from Paulish, “Architecture – Centric Software Project Management)

- | When a team member is struggling, others automatically help out
- | Team interactions are “comfortable”
- | Conflict is quickly resolved

- | Team member's know each other well and can anticipate reactions to proposed ideas
- | Team members are committed to the project

Characteristics of Good Teams (cont'd)



| Team members use “we” rather than “I” in discussing accomplishments and problems

| Team leader is viewed more as a “coach” than a “dictator”

| Team members understand their roles and responsibilities

| Team takes pride in its work

| Team learns from its mistakes

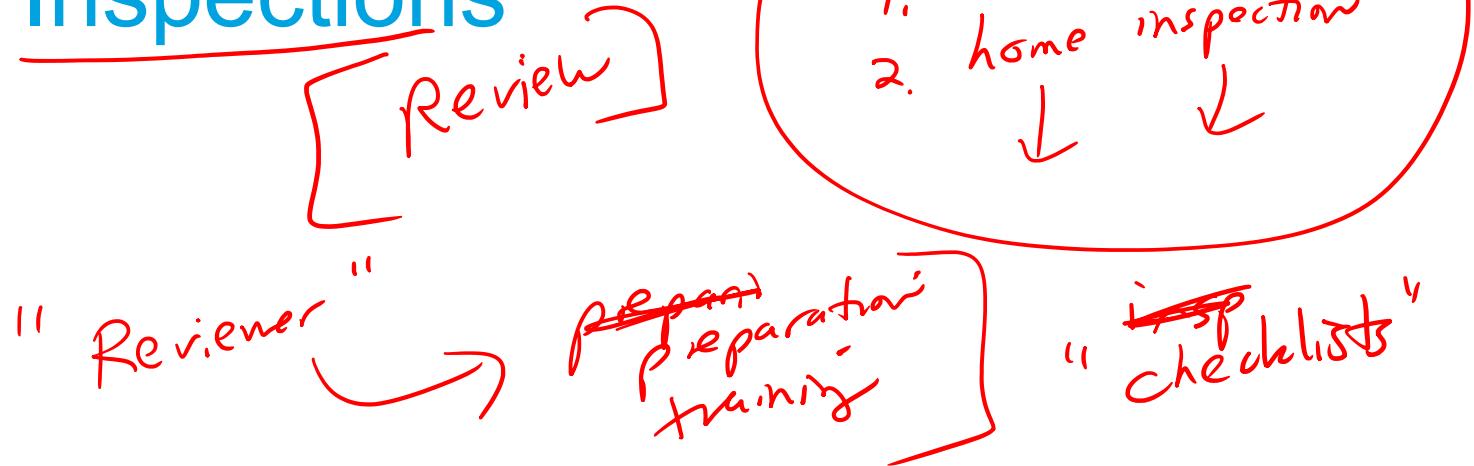
| team believes that it will be successful

Summary



Test Management Part 1

Test Inspections



Objective



Objective

Plan and
participate in
various work
product
inspections

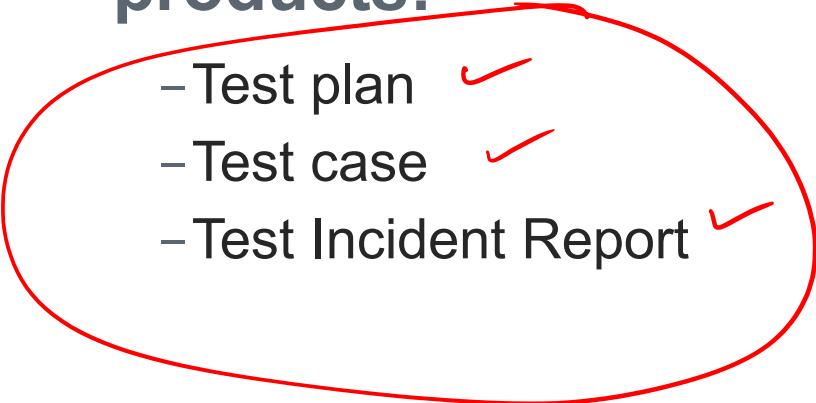
*test
deliverables*

System Tester Involvement in Inspections



| System testers should be involved in inspecting requirements documents during development

| System testers should also inspect their work products:

- Test plan
 - Test case
 - Test Incident Report
- 

Definitions

"Landiz"

P. 16 →
P.

FAA



| An inspection is a
“formal statistical
process control
method for evaluating
documents and their
production”

discipline

data

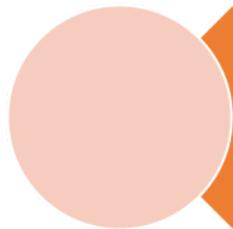
| Inspections require:

- Advanced preparation ✓
- Utilization of rules and checklists
- Metrics gathering and analysis to facilitate process improvement

Inspection Package



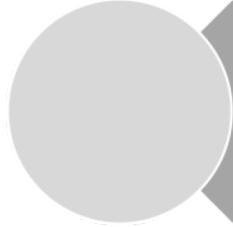
~~Test cases~~



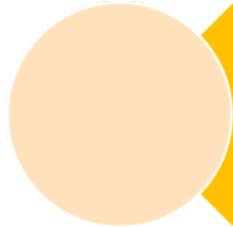
Work-product to be inspected



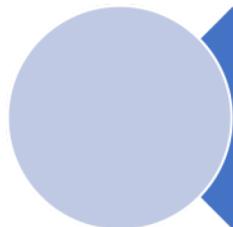
~~Req~~



Supporting documentation



Checklists



Inspection Agenda



Checklists

| "Job Aid" for inspectors

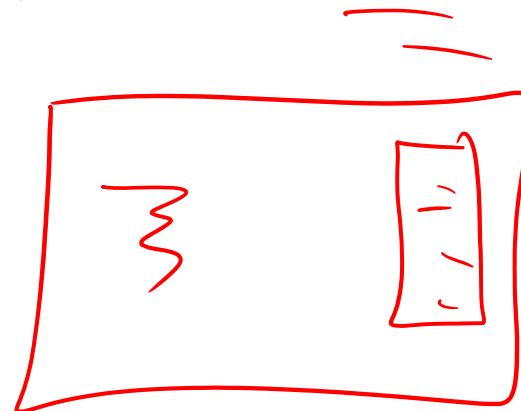


| Fundamental tool in the inspection process

| Guides the defect detection process

| Customized by project and document

| Available electronically



“Testability Concerns”

“code”

| Testers must advocate for **testability features**

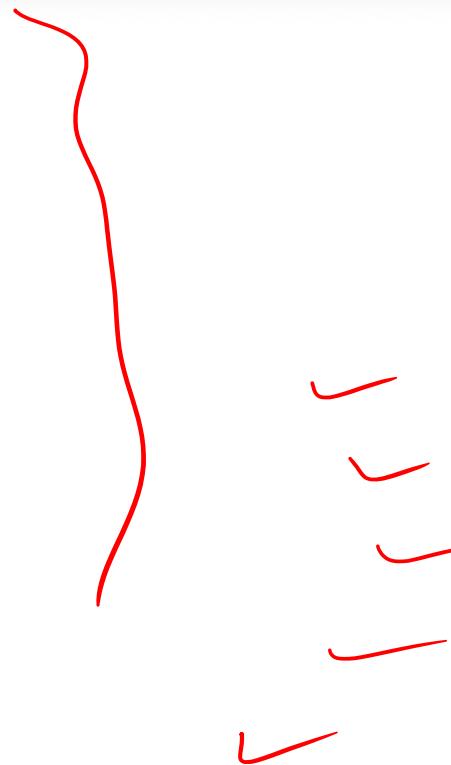
| Testability implies both visibility and control

| Examples include:

- Diagnostics ✓
- Test points to view or modify data during execution
- Ability to place the system into a given state ✓

Requirements Checklist

1. testability
2. Completeness
3. Correct
4. Consistent
5. clarity
- 6.
- 7.



Test Plan Checklist

1. test objectives defined / clear
2. test strategy
3. reasonable schedule
4. test environment defined
5. test priorities identified
- 6.
- 7.



Test Case Checklist

1.

repeatable ✓

2.

defined inputs / expected results

3.

environment / state ✓

4.

sequencing ✓

5.

6.

✓ ✓ ✓

7.

Bus documentation

- well defined

Summary



Test Management Part 2

Test Maturity Model

Objective



Objective

Assess the
maturity level of a
testing
organization.

Applicable Maturity Models



Capability Maturity Model

CMU - SEI



self
assessment

CMMI Levels

I - S

H M O S



y

| Managed: metrics are used extensively to guide process

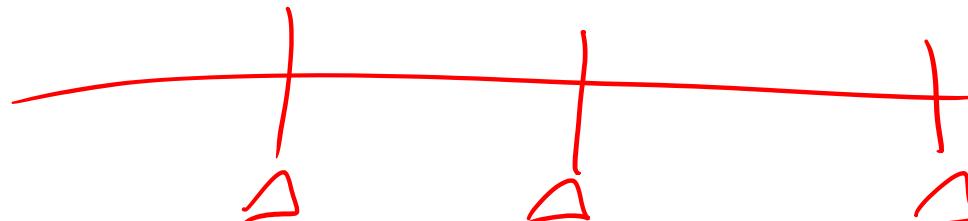
| Optimizing: emphasis on continuous improvement

| Initial: adhoc

| Repeatable: expertise lies in individuals

| Defined: processes are defined and documented

wf



- | Developed in an analogous manner to assess maturity of testing process
- | Five levels of maturity defined

| Each level has goals and subgoals that are achieved via ATRs (activities, tasks, responsibilities)

I - S

TMMI Level 1: Initial



| Ad hoc testing without specific goals

| Test process begins after code is written

| Testing is not a priority

| Test tracking is not performed

TMMI Level 2: Phase Definition

| Software testing plans are developed that include well defined testing phases

*unit
integration*

| Basic testing methods are introduced

| Goals for testing are defined

TMMI Level 3: Integration

- | Software testing is integrated into the software development process
- | Test progress is monitored and tracked

- | Testers are trained
- | Test plan risk management is performed

prioritize

TMMI Level 4: Management and Measurement



Metrics are introduced to assess test process

Review processes are introduced to assess effectiveness and efficiency

\$ / effort

effectiveness

~~detected~~ defects detected

defects detectable

TMMI Level 5: Optimization, Defect Prevention and Quality Control



| Root cause analysis is performed to prevent defects from reaching the customer

unit

| Statistical quality control used to monitor test process

| Test process improvement implemented

Summary

$$\underline{C_{MMI}} \approx T_{MMI}$$

Test Management Part 2

Test Outsourcing

Objective



Objective

Develop a plan for
outsourcing
testing.

Home Analogy



| **Do you do the work yourself or hire a contractor to do the work for you?**

Factors Influencing Outsourcing Decision



- | Need for specialized / advanced technology or capability
- | Strategic value of system
- | Cost
- | Strategic Alliance

- | Maintenance support
- | Speed of development
- | Desire for level staffing

Test Outsourcing Activities



- | Define the work to be subcontracted
- | Develop an outsourcing plan
- | Perform domain and/or process evaluations if needed
- | Select a subcontractor

- | Contract with subcontractor
- | Oversee the testing to ensure that it is on schedule, within cost, and meeting requirements
- | Accept the testing

Defining Subcontracted Work



| Partition the product and its activities

| Select subcontracted work to:

- Maximize organization's effectiveness
- Match skills and capabilities of potential subcontractors
- Minimize communication and coordination efforts
- Minimize dependencies
- Minimize risk of subcontractor knowledge not being transferred to the organization

Develop a Subcontractor Management Plan



| **Develop a technical specification for the work**

| **Develop a statement of work for the supplier**

- Identify all tasks to be performed
- Identify relevant processes to be followed
- Identify maintenance responsibilities

Develop a Subcontractor Management Plan (cont'd)



| Perform risk management

- Identify outsourcing risks early
- Adjust acquisition strategy to minimize risks

| Estimate resources needed

- Estimate supplier test effort (what it should cost)
- Estimate effort for vendor

Components of an Outsourcing Agreement



| Contract

| Technical specs

| Statement of work

Supplier Management Activities



| Selection

- Searching for and evaluating vendors
- Specifying legal terms of contract
- Negotiating contract

| Subcontract management

- Conveying and explaining requirements
- Monitoring vendor, including reviews
- Resolving problems

Factors to Consider in Selecting a Supplier



| Strategic business alliances

| Prior performance on similar work

| Geographic location

| Software testing and software management capabilities

| Available staff

| Domain expertise

| Cost

| Similarity of processes and tools

| Business viability

Contract with Supplier



| See lawyers!

Tracking and Oversight



| **A plan must be developed for supplier tracking and oversight**

| **Plan must address:**

- Activities to be performed and the schedule to perform them
- Identification of groups, assigned responsibilities and inter-group communication
- Techniques, tools and methodologies to be employed for review and tracking of vendor performance
- Escalation procedures

Tracking and Oversight Activities



| Maintaining good communication

| Continuous risk management

| Reviews

| Approval of invoices

| Metrics

Acceptance of Work



| Acceptance of work is a formal procedure which has contract implications

| You must be convinced that the testing is thorough and complete

Summary



Test Management Part 2

Test Process Improvement

Objective

→ faster
→ effectively



Objective

Explore
Strategies for
Improving the
Testing Process

Improving the Testing Process



| Software test teams
must constantly look
for ways of testing
faster and more
effectively

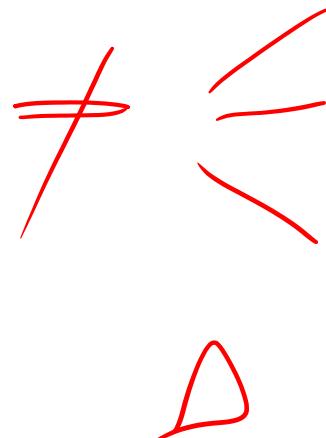
"ASU
Football"

| Process improvement phases

- Characterize the current process
- Analyze current process
- Characterize target process
- Process redesign
- Implement

Characterizing the Current Process

| It is important to understand the current testing process



| **Care must be exercised in distinguishing among:**

- The Perceived Process: What you think you do.
- The Official Process: What you are supposed to do.
- The Actual Process: What you do

Quote from Watts Humphrey

SEI "CMM"

"If you don't know where you are, a map won't help."

Analyze Current Process

| Identify value-added and non-value added activities

"*Analytics*"

| Software metrics play a key role in objective process analysis

| Identify the sources of variation in the process



| Software test leader must plan to collect the appropriate metrics

Testing Metrics

"Test"

||

| You cannot control
what you cannot
measure ||

| GQM paradigm (goal-
question-metric)

- ✓ – Define the goals of the measurement process ✓
- ✓ – Derive the questions that must be answered to meet the goals
- Develop metrics to answer the questions

Example Test Goal

||

||

||

||

| Reduce testing time

| Find more severe
defects

"Reduce APS utility bill during
the summer" KWH

- How much energy?
- When is it used?
- Where?

Example Testing Questions



| Where do we spend most of our time during testing?

| What types of defects are we missing?

| What is our testing productivity
=====

Possible Testing Measures



| Test productivity

| Test quality

EQM

Characterize Target Process



| Identify the ultimate goal of the process

e.g.,

- Minimize test time
- Find more high severity problems

| Quote from Watts Humphrey

- “If you don’t know where you are going, any road will do.”

GQM

"Process Redesign"

| Improve the current process in the direction of the target process

| Explore ways for eliminating, ✓ simplifying or combining activities

| Explore ways of eliminating rework

| Explore ways of reducing task variance

"priority"

Unit test



Implement Process Improvements



| Set process improvement goals

| Implement process improvements

- Begin with candidates that are well defined
- Pilot candidates that are not proven

Measure progress towards goals

Implen

Process Improvement Tools - Post Mortem / Lessons Learned / Retrospectives

A set of lessons learned is documented

project
review test

"PM"

Analysis techniques include:

- Interviews with key personnel
- Statistical analysis of data
- Investigations of major problems
- Identification of what went well and what didn't go well

Testing Lessons Learned Addresses



| Overall schedule
analysis

| Adequacy of entrance
criteria

| Adequacy of exit
criteria

| Overall quality



| Team interactions



Example Questions to Address



- | How effective was communication and collaboration with development?
- | How effective was test estimation?

Are there any issues with the test environment?

What types of defects are we missing?

Where are we wasting time?



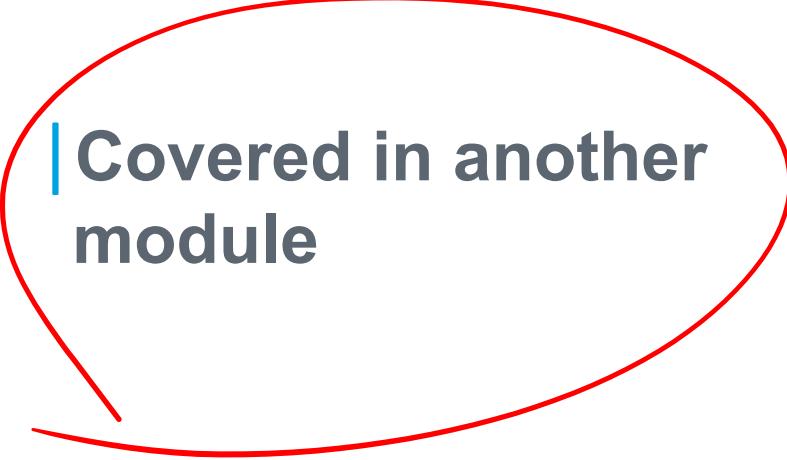
Process Improvement Tools - Causal Analysis



A decorative horizontal bar at the top of the slide, consisting of several colored segments (yellow, blue, orange) separated by thin black lines. Red hand-drawn style lines highlight the first yellow segment, the first blue segment, and the first orange segment from the left.

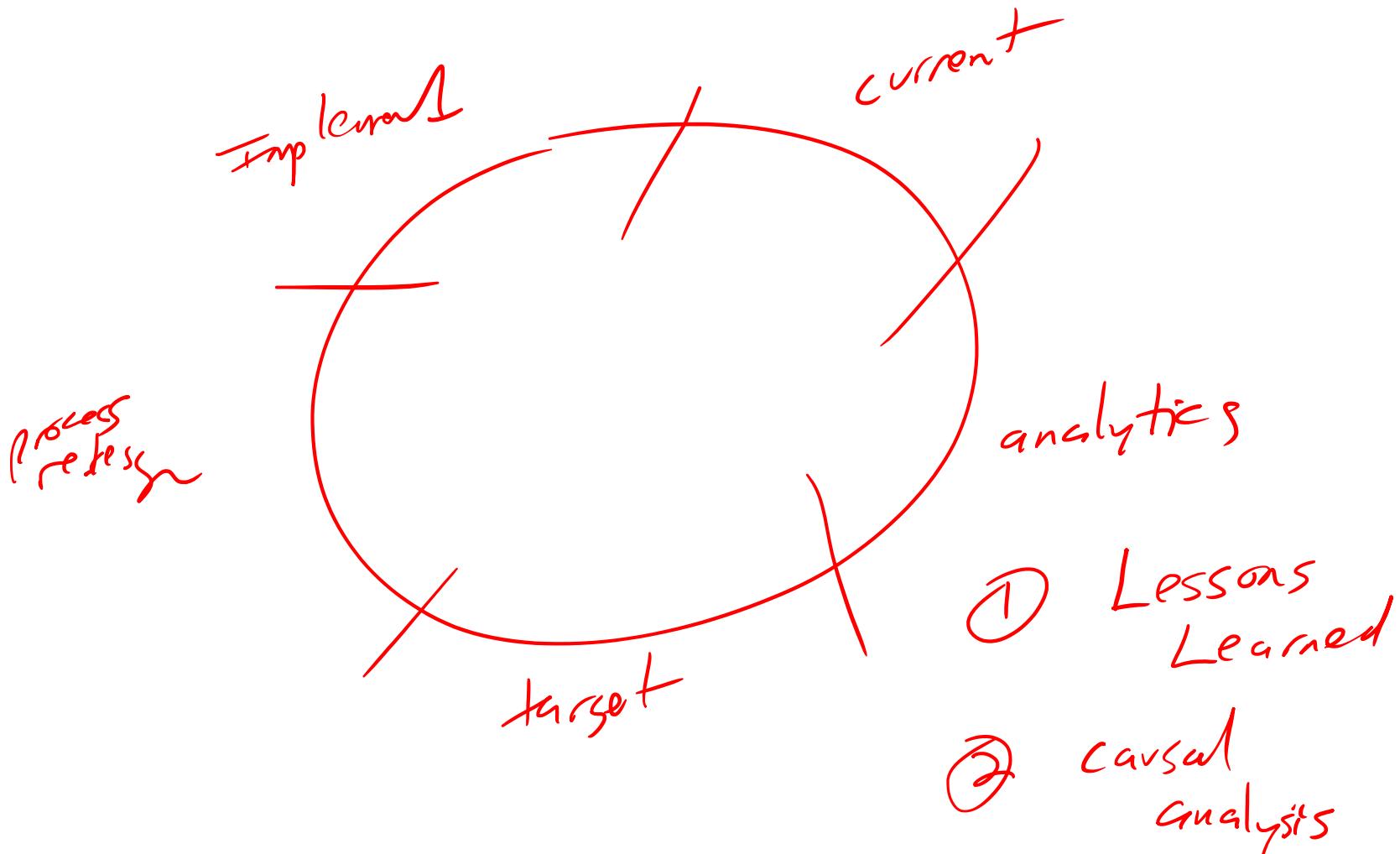
| Seeks to identify the root cause of defects as well as approaches to eliminate future occurrences

Covered in another module



A large red oval is drawn around the text "Covered in another module". A red arrow points from the bottom-left towards the bottom of the oval.

Summary



Test Management Part 2

Test Tracking

Objective



Objective

Utilize various
measures to track
testing progress.

System Test Tracking



| During system test information must be collected and tracked to assess:

- Product quality (previously discussed under "system test exit criteria")
- Testing progress

| Numerous measures exist for tracking test progress against a plan including:

- Percentage of tests developed
- Percentage of tests executed
- Percentage of requirements tested

| Testing schedule and effort progress can also be assessed via earned values

Earned Values



| **Earned values are a technique for tracking both schedule and cost progress**

| **Earned value approach establishes a relative value for every task and credits that value when the task is completed**

| **Progress is then tracked in terms of:**

- BCWS (Budgeted Cost of Work Scheduled)
- BCWP (Budgeted Cost of Work Performed)
- ACWP (Actual Cost of Work Performed)

Earned Value Example



Assume that we are concerned with testing two concurrently developed increments of a product labeled "1" and "2" and for each increment there are features labeled "A" - "F". Thus, task "1A" corresponds to the activity of testing feature "A" in increment "1". Associated with each testing task is an earned value corresponding to the estimated effort to complete the task.

Tasks	EV	Tasks	EV
1A	50	2A	30
1B	40	2B	40
1C	30	2C	30
1D	20	2D	50
1E	50	2E	40
1F	30		

Earned Values Example

(continued)

Assume the following schedule:

Week 1	Week 2	Week 3	Week 4
1A, 1B	1C, 1D	1E, 1F	
	2A, 2B	2C, 2D	2E

Earned Values Example

(continued)

Week	Work Completed	Cost
1	1A, 1B, 1C	100
2	1D, 2A, 1E	70

Earned Values Example

(continued)

	Week 1	Week 2	Week 3	Week 4
BCWS	90	210	370	410
BCWP	120	220		
ACWP	100	170		

Earned Values Example

(continued)



| Based on earned values the project is ahead of schedule at the end of week 2

BCWS = 210

BCWP = 220

| Based on earned values the project is below budget at the end of week 2

BCWP = 220

ACWP = 170

Summary

