
# Stewart-Priven Group, LLC

# Fixing Software Inspections

## Washington DC – Users Group

**July 22, 2008**

7962 Old Georgetown Road, Suite B, Bethesda MD 20814
865-458-6685 ï Fax: 865-458-9139 ï SPGroup@charter.net
www.stewart-priven.com

# Are Inspections Broken?

## What Are Inspections?

- Software Inspections are a pre-emptive peer review of work products
  - By 3-5 trained individuals (e.g., stakeholders) using a well defined (formal) process
  - To detect and eliminate defects as early as possible in the Software Development Life Cycle, or closest to the points of injection

Commonly inspected work products include:
- Requirements
- Design
- Interface Agreements
- Code, Test Material, or other text

- Referred to as "Peer Reviews" in Capability Maturity Model (CMM) and CMMI

# Stewart- Priven Background

**30 years software development experience (each)**

- Commercial, Executive Mgmt. Focus
- Government, Technical/Project Mgmt. Focus

**Managed IBM team that developed Inspections (led by M. Fagan)**

**Both taught Inspections for M. Fagan 1998 – 2005**

- 50 company locations, 250 classes, 5000 practitioners

**SPG, LLC formed Feb. 2006**

- Systems & Software Technology Conference, **Tutorial** June 2007 - Tampa
- Crosstalk Magazine **Article** January 2008
- Practical Software Quality & Testing, Two **Presentations** May 2008 - Las Vegas
- International Conference - SW Process Improvement: October 2008 Wash. DC
  - **Tutorial Topic**: ñNew Direction for the Inspection (Peer Review) Processò
  - **Presentation Topic**: ðManagement Responsibilities for Successful Inspectionsò

# Concerns of Other Industry Experts

- "Inspection . . .depressingly few practitioners know about the 30 year old technique of software inspection. Even fewer routinely perform effective inspections or other types of peer reviews." *Karl Wiegers - "The More Things Change, Better Software, Oct. 2006"*

- 'Formal inspections . . .. part of the problem here is that not a lot of companies know how to use these things.' *Capers Jones, Chief Scientist, SPR – "Computer Aid Inc. July 2005"*

- "Yet for others, Inspections never succeeded as well as expected, primarily because these organizations did not learn how to make Inspection both effective and low cost." *Ron Radice - "High Quality Low Cost Software Inspections, 2002 Paradoxicon Publishing"*

- "Inspections . . .with the right instrumentation and training they are one of the most powerful techniques for defect detection." *Chris Ebert - "IEEE Software Jan/Feb 2000,* ***Best Influences on Software Engineering over past 50 years"***

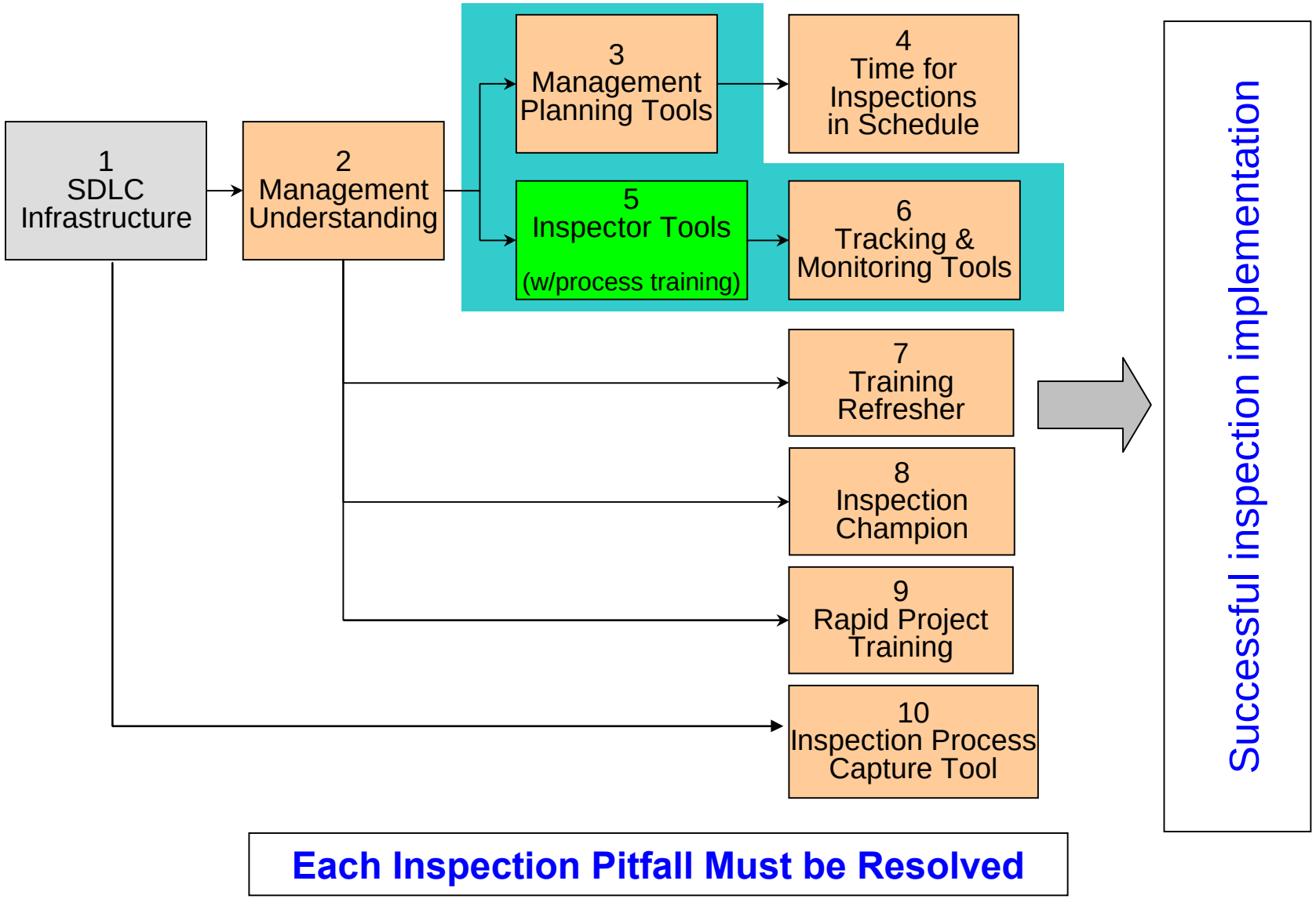
# Inspection Pitfalls

***Pitfalls cause inspections to fail! Each must be Resolved***

1. Lack of supportive Development Life-Cycle infrastructure for Inspections
2. Lack of understanding by upper managers of the process, benefits, their role
3. No Management tools for up-front planning, and estimating potential savings
4. No Time incorporated in development schedules for performing inspections; and
   No Criteria established for selecting quality-critical areas to inspect
5. No Inspector tools for correct & consistent execution, assessment, & reporting
6. Limited Monitoring of inspection execution and results
7. No post-class student refresher
8. Lack of Company Champion to facilitate consistent implementation
9. Too much elapsed time occurs between training of project team members
10. No Company Inspection Process Capture mechanism

Stewart, Roger & Priven, Lew. ñHow to Avoid Software Inspection Failure and Achieve Ongoing Benefits.ò
CROSSTALK Magazine Jan. 2008 (or http://www.stsc.hill.af.mil/crosstalk/2008/01/index.html)

# Inspection Pitfall Roadmap



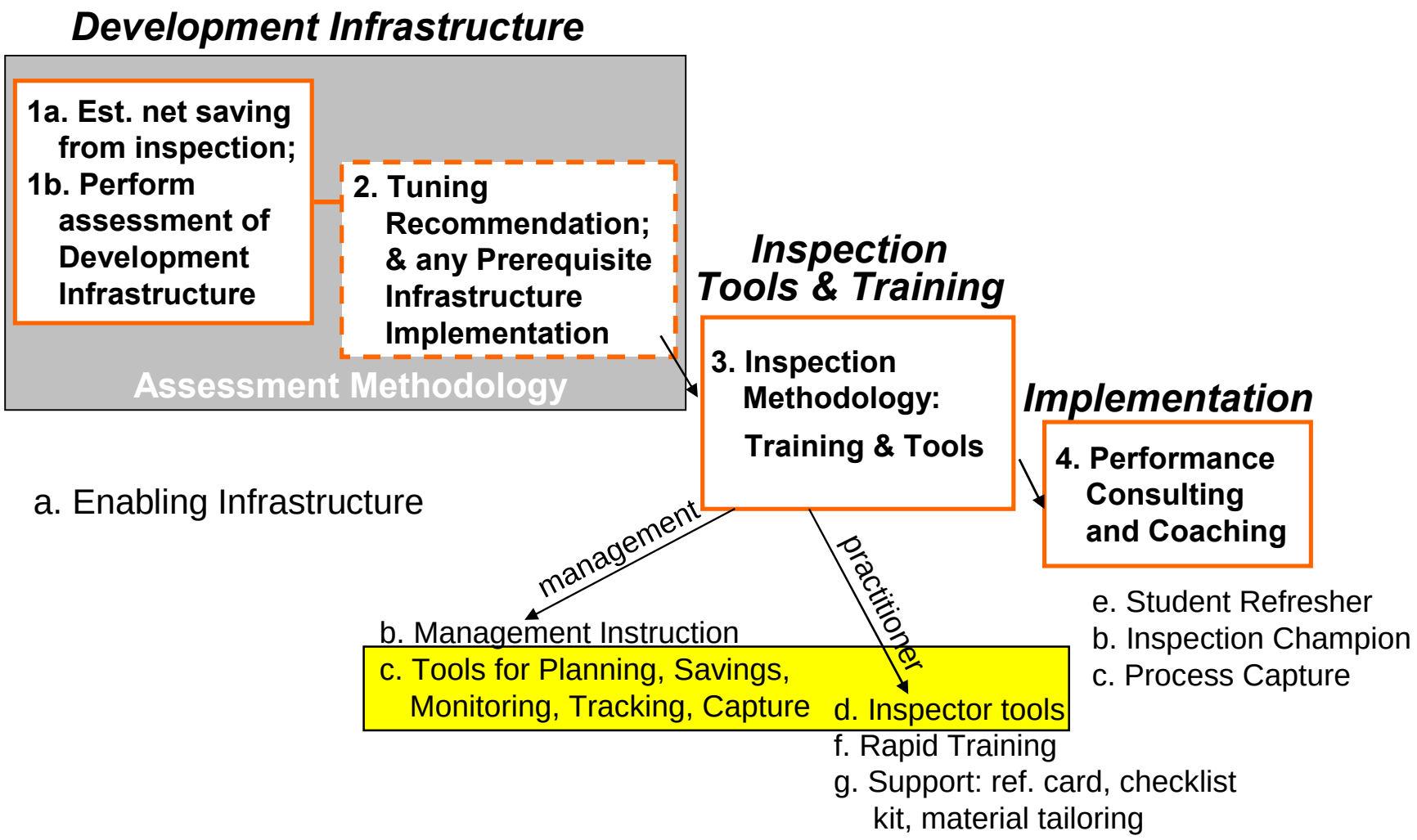

**Each Inspection Pitfall Must be Resolved**

# Resolving Inspection Pitfalls

| **SOLUTIONS to Fixing SW Inspections** | PITFALL #s |
|---|---|
| **a. Pre-Requisite Assessment of client SDLC** | 1 |
| **b. Upper management class in 'Managing Inspections'** | 2, 4, 8, 9 |
| **c. Tools for Management Planning, Savings, Monitoring, Tracking, & Process Capture** | 3, 4, 6, 10 |
| **d. Inspector Tools (Preparation, Execution, Analysis)** | 5 |
| **e. Student Refresher** | 7 |
| **f. Rapid training capability (<= 2 days) & Mgmt. Commitment** | 9 |
| **g. Inspector Support**<br>**- Student class feedback to Management**<br>**- Inspection Role Reference Card**<br>**- Product Inspection Checklist Kit**<br>**- Course material tailoring** | 7 |

**A foundation for:** ***AN INSPECTION 'METHODOLOGY'!***

# Using an Inspection 'Methodology'



# Inspection Tools – SPG's *SpectorTool Suite*™

Planning Tools
(Project Planers)

Execution & Analysis Tools
(Inspection Leader, with team)

Tracking & Monitoring Tools
(Managers)

**PlanSpector ™** ***Before any Inspections***

- ***Supports <u>Project Planners</u> acquiring and providing senior management with cost savings data to support using inspections***

**InSpector ™** ***During Inspection***

- ***Supports <u>Inspection team</u> finding maximum defects through proper inspection process execution***

**TrackSpector ™** ***After Inspection***

- ***Supports <u>Management</u> with immediate results, savings and ROI of individual and multiple inspections***

# Stewart-Priven Inspection Tools

Planning Tools
(Project Planers)

Execution & Analysis Tools
(Inspection Leader with team)

Tracking & Monitoring Tools
(Managers)

***Before any Inspections*** | ***During Inspection*** | ***After Inspection***

***Tool Use Sequence***

PlanSpector™

- Inspection Planning Counter — 1
- Project Savings/Cost Estimator — 2

InSpector™

- Inspection Preparation Tool — 3
- Inspection Meeting Log Tool — 4
- Analysis Tool — 5

TrackSpector™

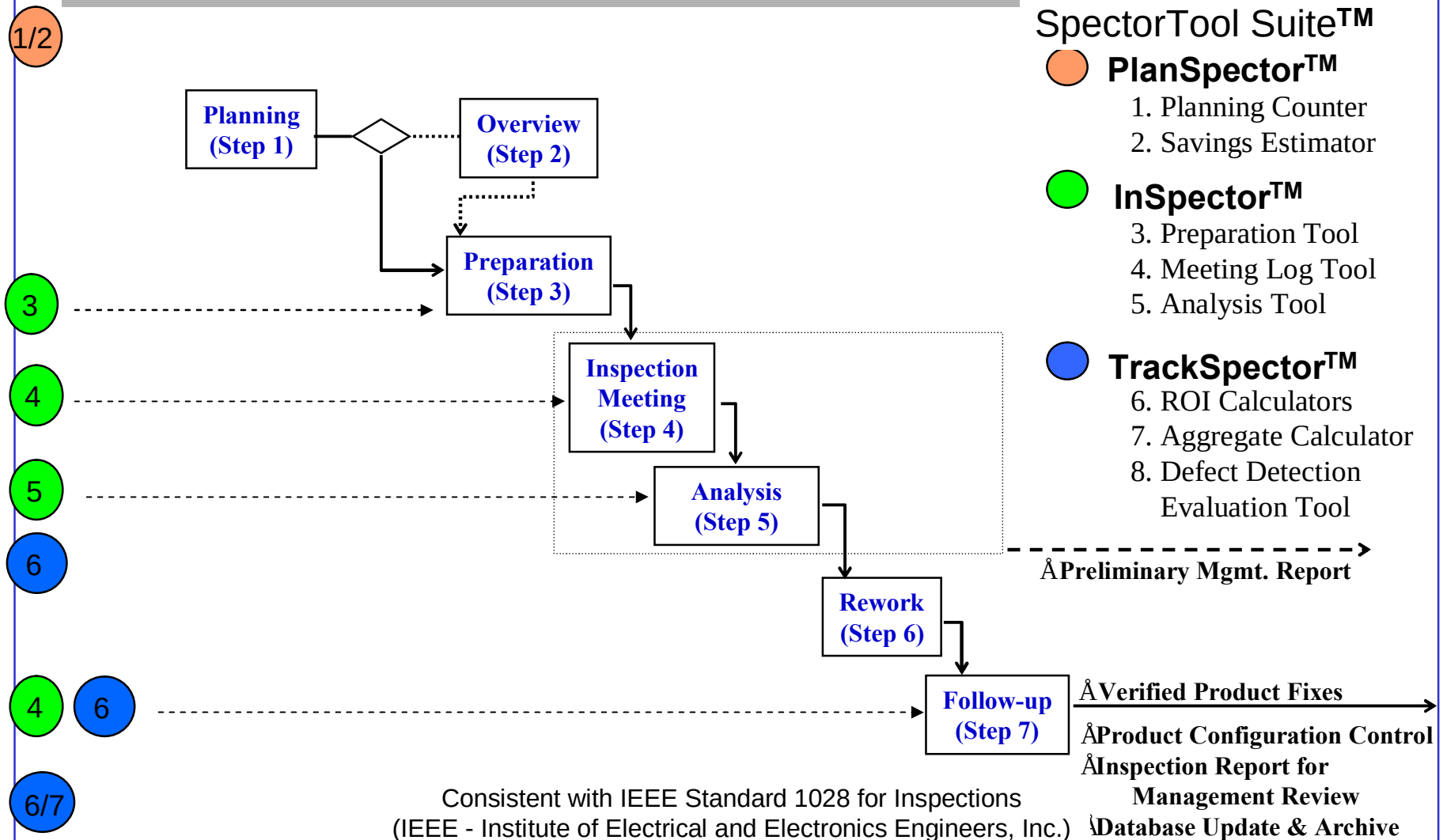
- ROI Calculators -Requirements -Design -Code — 6
- Aggregate Results Calculator — 7

Defect Detection Evaluation Tool — 8

Insp | Test | other

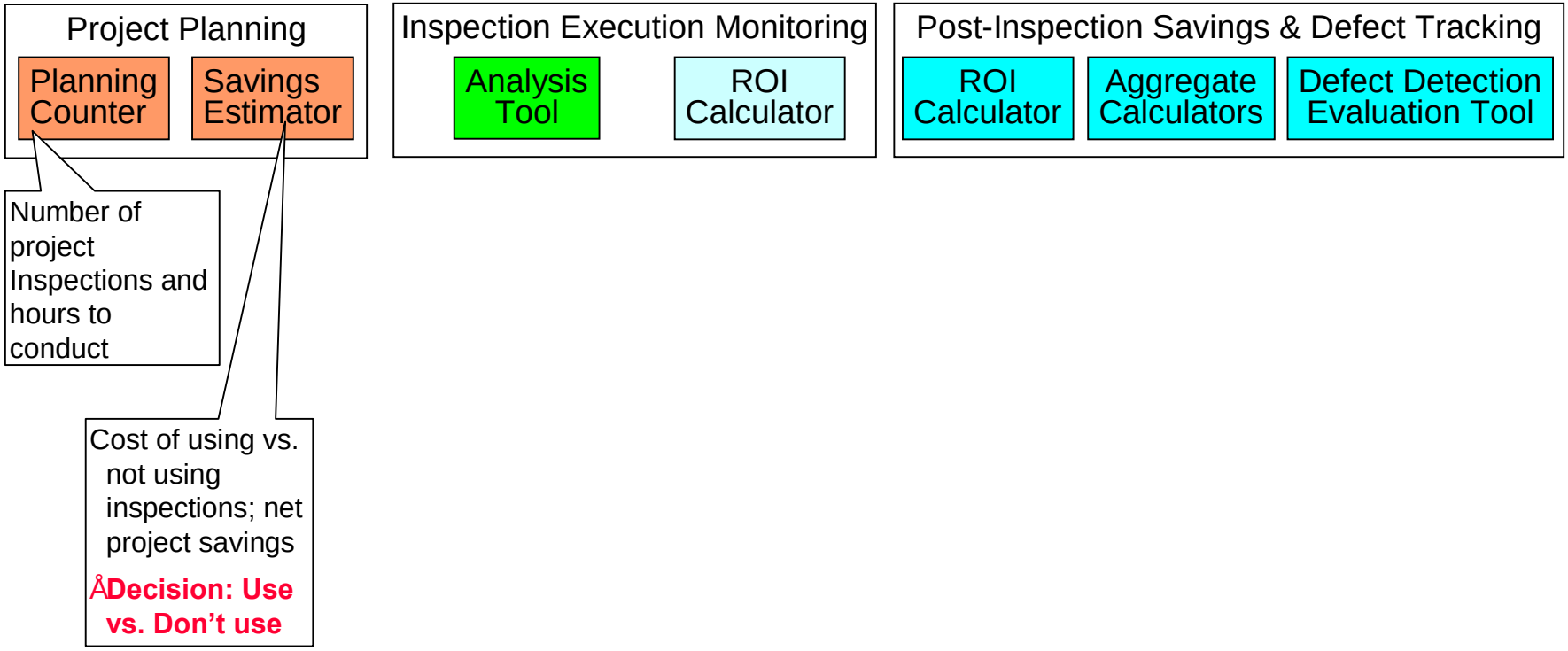
# Inspection Steps: Objective, Find & Fix Product Defects

## *Where are Inspection Tools Used?*

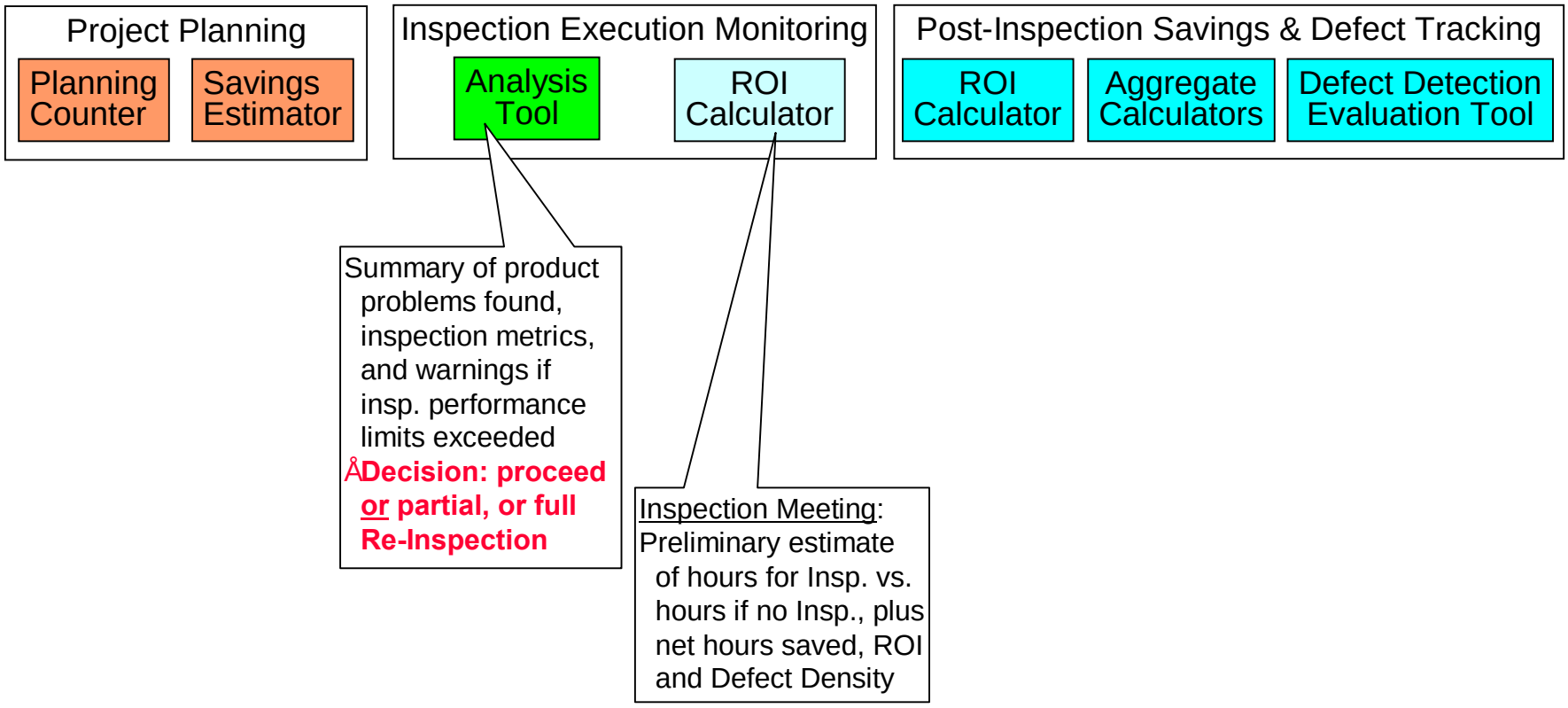



Consistent with IEEE Standard 1028 for Inspections
(IEEE - Institute of Electrical and Electronics Engineers, Inc.)

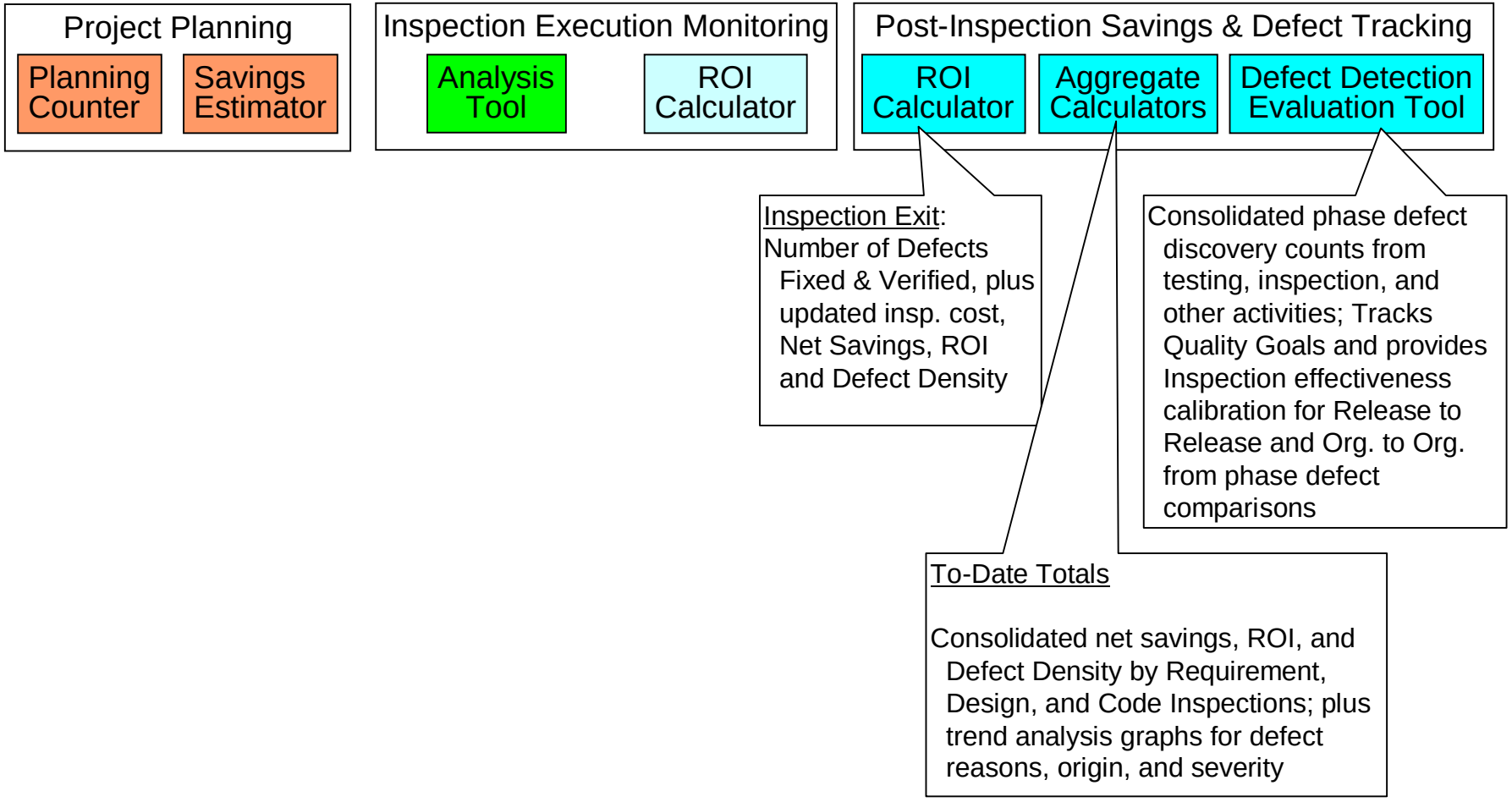
# Planning Inspection Tools for Management



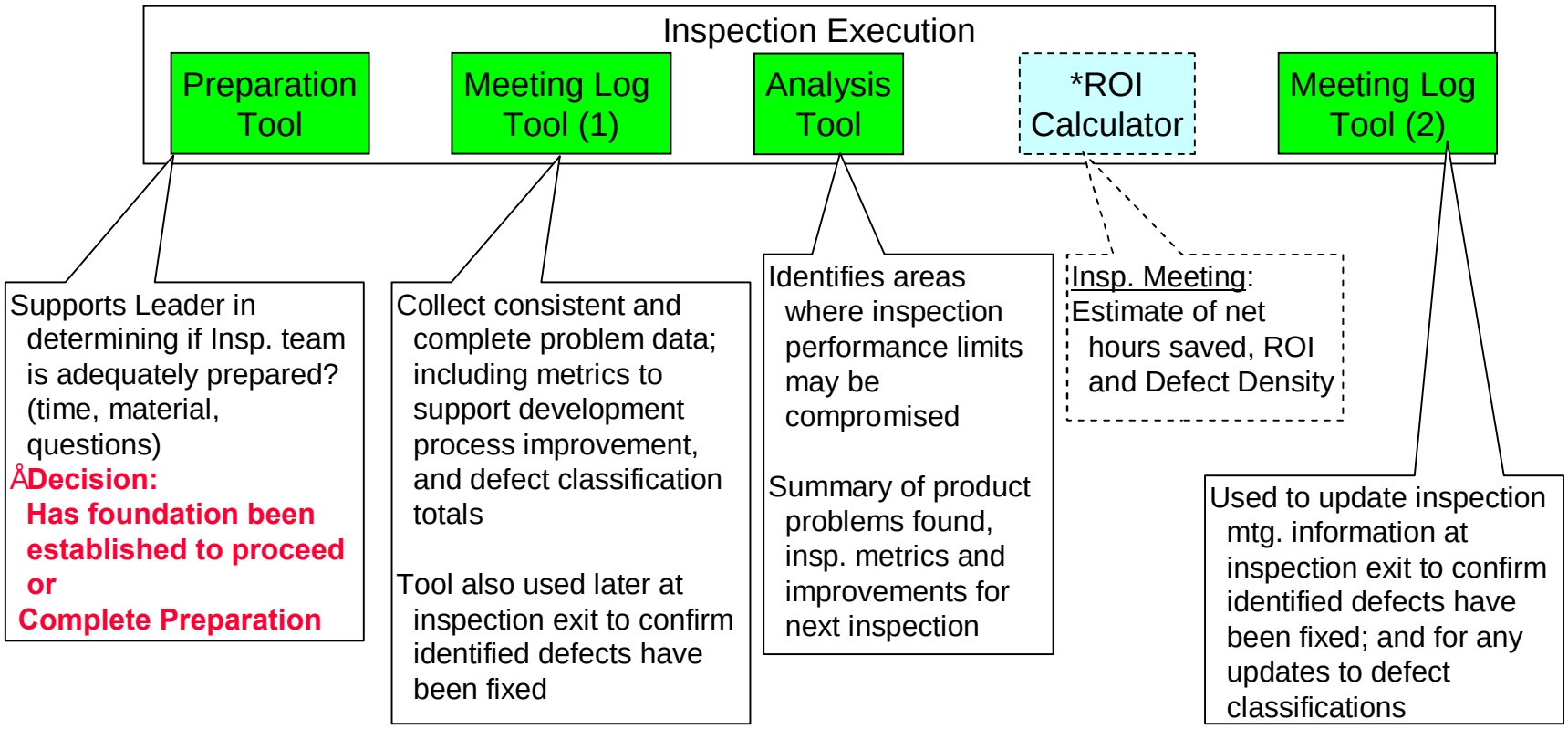
# Monitoring Inspection Tools for Management



# Tracking Inspection Tools for Management



# Practitioner Inspection Tools

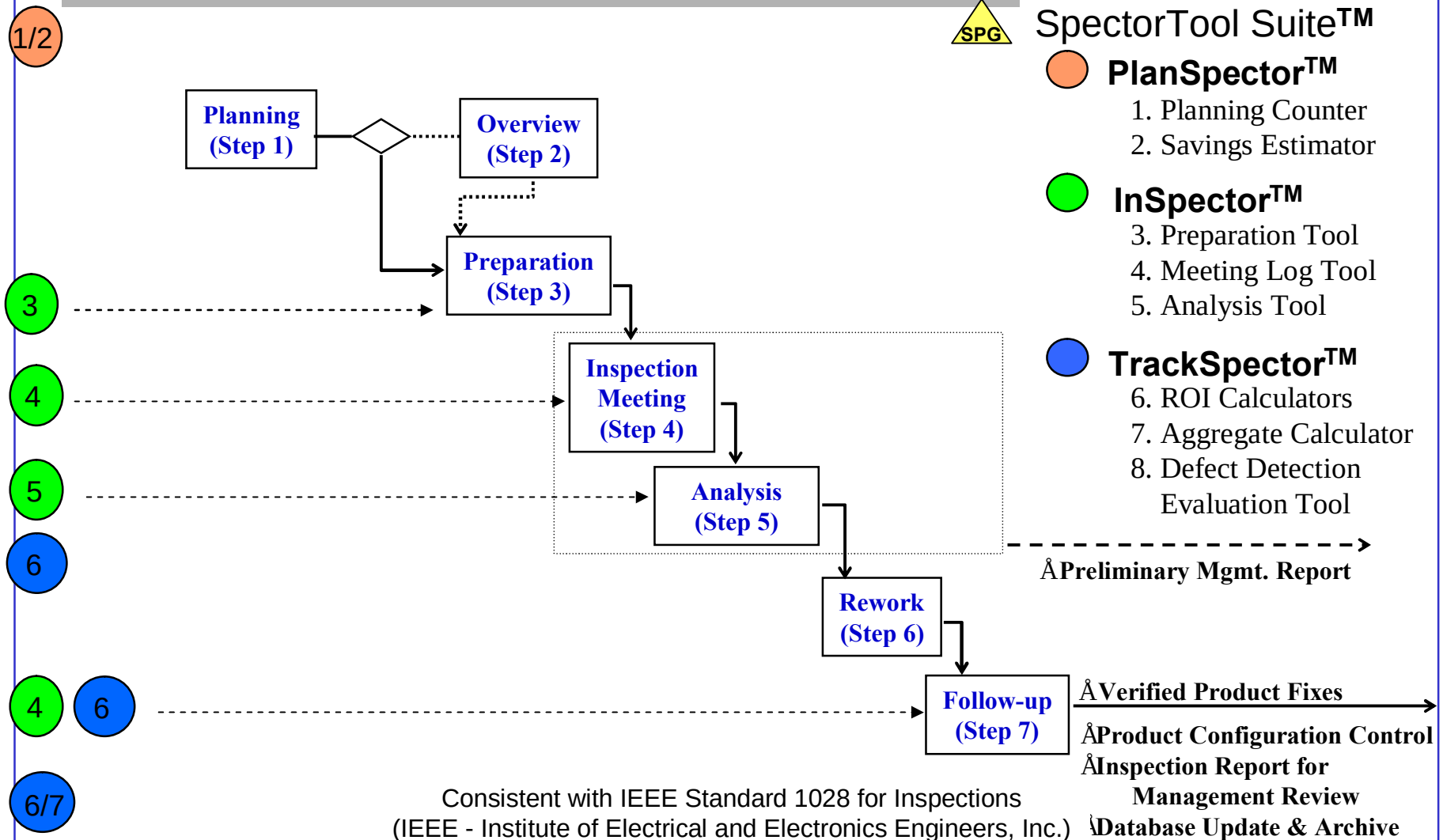



* Primarily a management tool, but is available to inspection team during Analysis step

InSpector™ TrackSpector™

# Inspection Tool Wrap-up

*Where are Inspection Tools Used?*




Consistent with IEEE Standard 1028 for Inspections
(IEEE - Institute of Electrical and Electronics Engineers, Inc.)

# AN IMPROVED QUALITY (and Cost) PROCESS

## Integrating an Inspection Methodology into the Software Development Life Cycle

Before Development

During Development

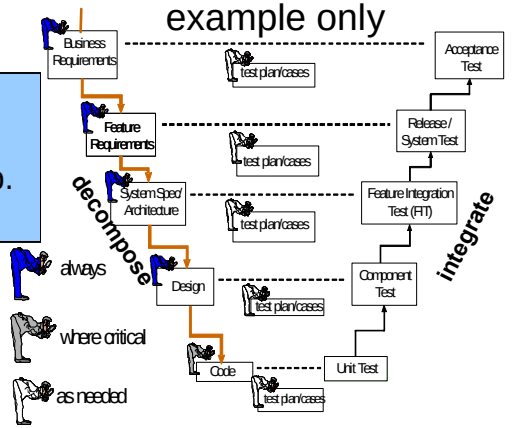
Estimate net project cost savings from inspections

Ensure development infrastructure is ready for inspections

Make necessary changes to infrastructure to handle inspections

ÅTrain Staff in Inspection Process
ÅTrain Mgmt. in Insp. Responsibilities

example only




Inspection Execution Support & Implementation Review

ÅCollect Individual inspection data

ÅGenerate Management inspection reports (interim & final)

Setting & Measuring Quality Goals for earlier Defect Detection

Potentials

History

Gather:
ÅDefect Potentials
ÅDetection History
ÅSet Quality Goals

Gather:
ÅAggregate Inspection data
ÅCollect Test Defect Data

ÅReview Interim Defect Detection Reports
ÅMake Process Adjustments
OR
ÅProceed

ÅReview Final Defect Reports
ÅMake Process Adjustments
ÅSet new quality goals

Inspection Methodology

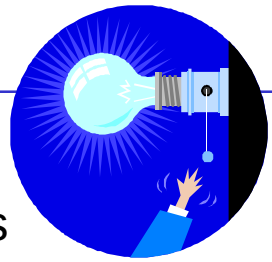
Developer

Before Development

During Development

After Development

# Fixing Software Inspections - Wrap-Up



**Use an Inspection <u>Methodology</u> that ensures . . . .**

- Projectôs infrastructure is ready for, and can support, inspections
- Management understands the process, itsôbenefits and their responsibility
- Computerized tools for project inspection planning and savings estimation
- Computerized tools for Inspectorôs correct and consistent execution
- Computerized tools for management inspection monitoring & results tracking
- Rapid training and student refreshers provided, along with ongoing inspection champion support
- Tool use repeatability for Inspection conduct-continuity and metric collection across teams, organizations, projects, locations, vendors
- Measurement of actual defect detection to ensure project quality goals for early defect removal are being achieved

**an ‘Inspection Methodology’ provides <u>much more</u> than an ‘Inspection Process’**

# Stewart-Priven Group - Contact Information


Address

Stewart-Priven Group, LLC
7962 Old Georgetown Road, Suite B
Bethesda MD 20814 USA

Phone

865-458-6685

Fax

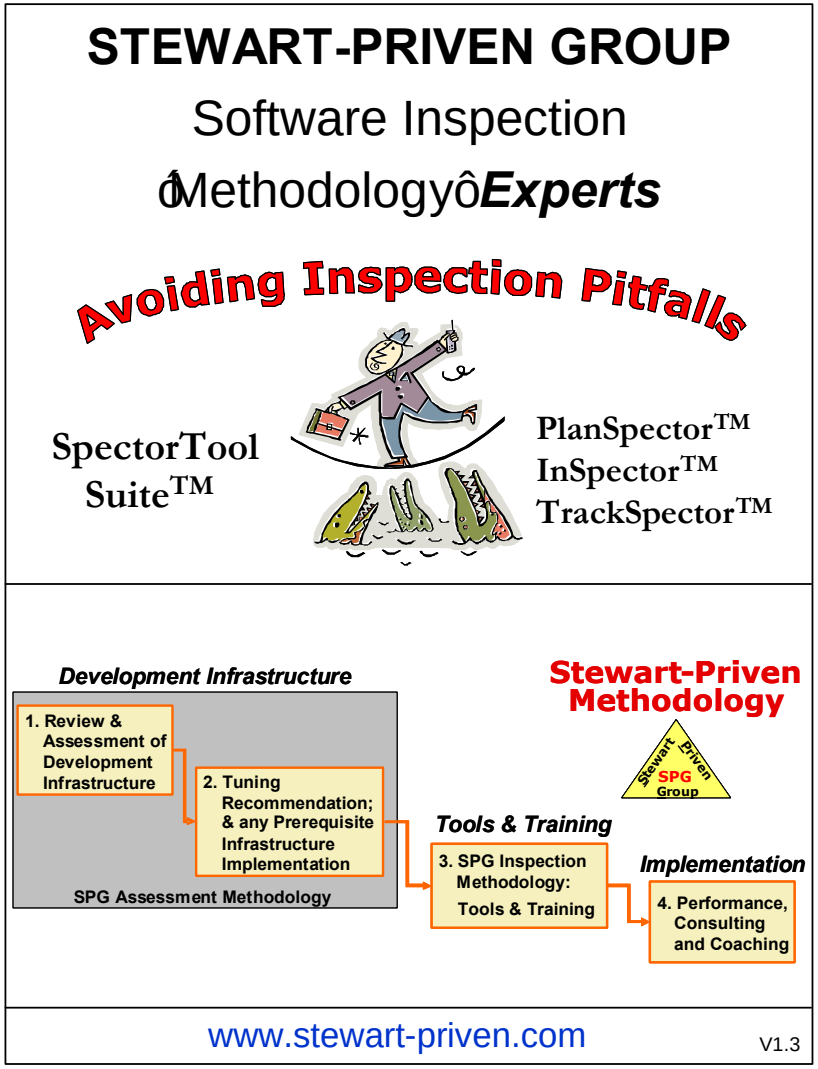
865-458-9139

Email

spgroup@charter.net

Web Site

www.stewart-priven.com

# Reference Material

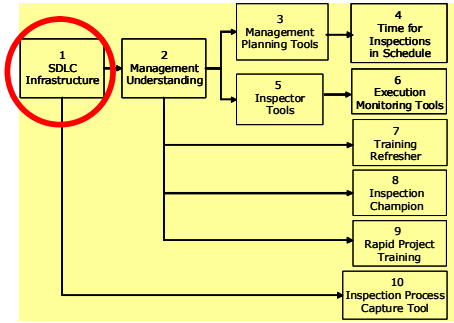
# Additional Benefits of Inspections

- **Use of inspection tools supports continuity & metrics across teams, organizations, projects, locations, companies**
- **Inspections are the fastest way to learn new products/domains**
- Consistent adoption of software and text standards
- Early identification of defect-prone work products

- Avoid repeating defects by providing rapid feedback to the developers
- Sharing of good ideas and identification of less effective approaches
- *Reduce defect injection by optional identification and removal of process defects*

# Inspection Pitfall #1 – Development Infrastructure

## Lack of supportive Software Development Life Cycle infrastructure




Refer to ôRoadmapô chart

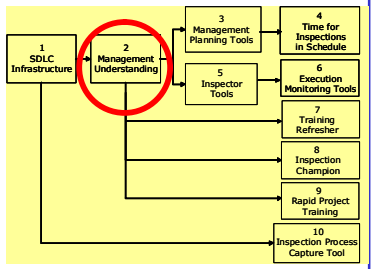
Pitfall Risks:

- Successful Inspections require skill in:
  - Planning
  - Data Collection
  - Reporting
  - Monitoring
  - Measurement Tracking
- Cultures without development infrastructures encompassing these skills are fraught with risk!
  - Critical inspection deviations introduced
  - Findable defects escape to later more costly phases

CMM®, CMMI® introduce ôPeer Reviewsô at level 3

# Inspection Pitfall #2 – Management Understanding

Lack of understanding by upper management and project leaders of the Inspection process, itsôbenefits, and their role

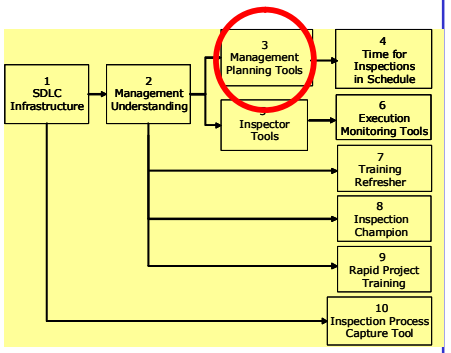



Pitfall Risks:

- Slow or incomplete Project team training and implementation
- Inadequate schedule, or no time allocated to inspections
- Inspection execution not monitored; results not tracked
- No óChampionô(focal point) to facilitate Inspection implementation

# Inspection Pitfall #3 – Savings Estimation

No computerized management tools for up-front inspection planning and estimating savings

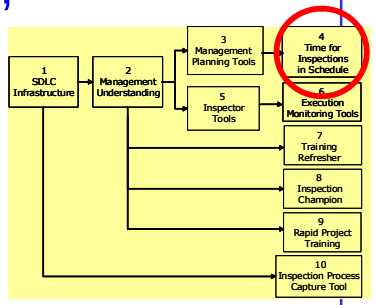



Pitfall Risks:

- Not understanding projectôs cost impact of not doing inspections
  - Tools not available to accurately estimate:
    - Cost of performing inspections
    - Net project savings of performing inspections
- Results in no inspections or not enough inspections

# Inspection Pitfall #4 – Time in Schedule

No time in development schedules for performing Inspections;
No criteria for selecting quality-critical areas to inspect



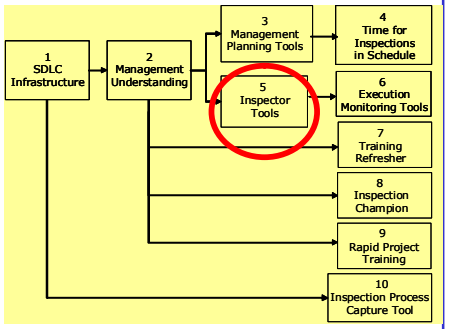

Pitfall Risks:

- Life-Cycle view of investing early to reap benefits later not understood, or practiced
  - Inspections not a stand-alone quality process
- Cost and Schedule impacts primarily borne by up-front requirements, design and implementation groups
  - while benefits accrue to later testing phases and customer use
- Inspections rushed, not performed correctly, or not done
- Malicious Compliance emerges - check the box - go through the motions
- Findable defects escape to later more costly phases

**Management at all levels in each development phase must support**

# Inspection Pitfall #5 – Inspection Execution

No computerized inspector tools to aid consistent and correct execution, analysis, and reporting inspection results

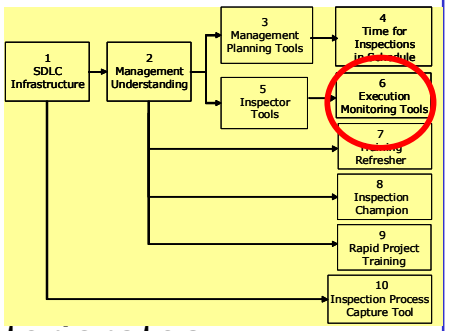



Pitfall Risk:

- Inconsistency and compromising shortcuts take hold
- No reinforcement, or feedback on improving inspection execution
- Management difficulty monitoring inconsistent inspection execution
- Findable defects escape to later more costly phases

# Inspection Pitfall #6 – Monitoring & Tracking Results

Inadequate management monitoring and tracking of inspection execution and inspection results

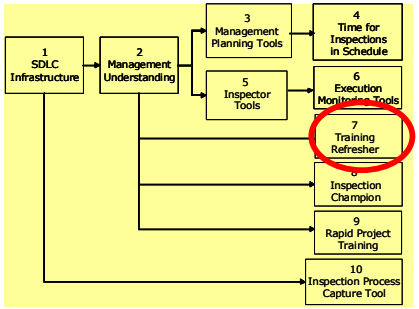



Pitfall Risk:

- Inspection process adherence and consistent execution deteriorates
- Findable defects escape to later more costly phases
- Management doesn't see inspection ROI, or defect-prone areas
- Employees eventually lose interest when benefits not periodically shared

# Inspection Pitfall #7 – No Refresher

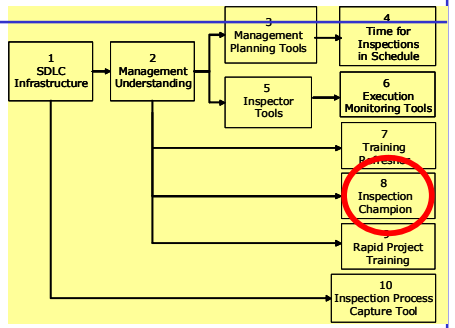
No post-class student refresher




Pitfall Risks:

- No reinforcement of key inspection principles
- Process misunderstandings; compromising shortcuts introduced
- Post-class situations from project inspections not handled effectively
  - Employees incorrectly conclude inspections not compatible with their environment
- Findable defects escape to later more costly phases
- Inspection implementation doesn't complete or not fully deployed

# Inspection Pitfall #8 – No Champion



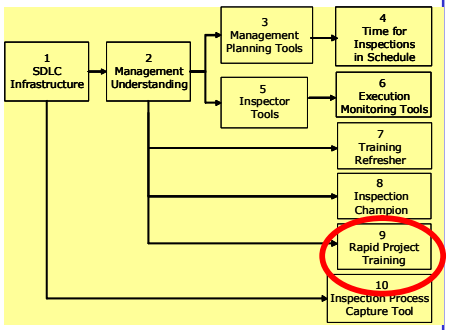

## Lack of Company (Project) Champion to facilitate consistent Inspection implementation

Pitfall Risks:

- Issues, situations, and resolutions not addressed or coordinated across other affected development areas or project teams
- Inconsistent and incorrect Inspection execution on same project
- Other projects implement Inspections differently
  - results not comparable or able to be consolidated (rolled-up)
- No timely support ï findable defects escape to later more costly phases
- Little óvaluableôfeedback to management

# Inspection Pitfall #9 – Extended Training

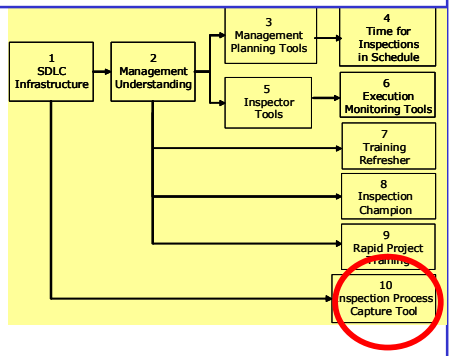
Training period for project teams is too long




Pitfall Risks:

- Ineffective start – too many untrained inspectors
  - Incorrect execution
- Acquired training skills deteriorate waiting for more trained inspectors
- Inspections never start – waiting on more people to be trained
- Findable defects continue to escape to testing, or worse

# Inspection Pitfall #10 – Process Capture

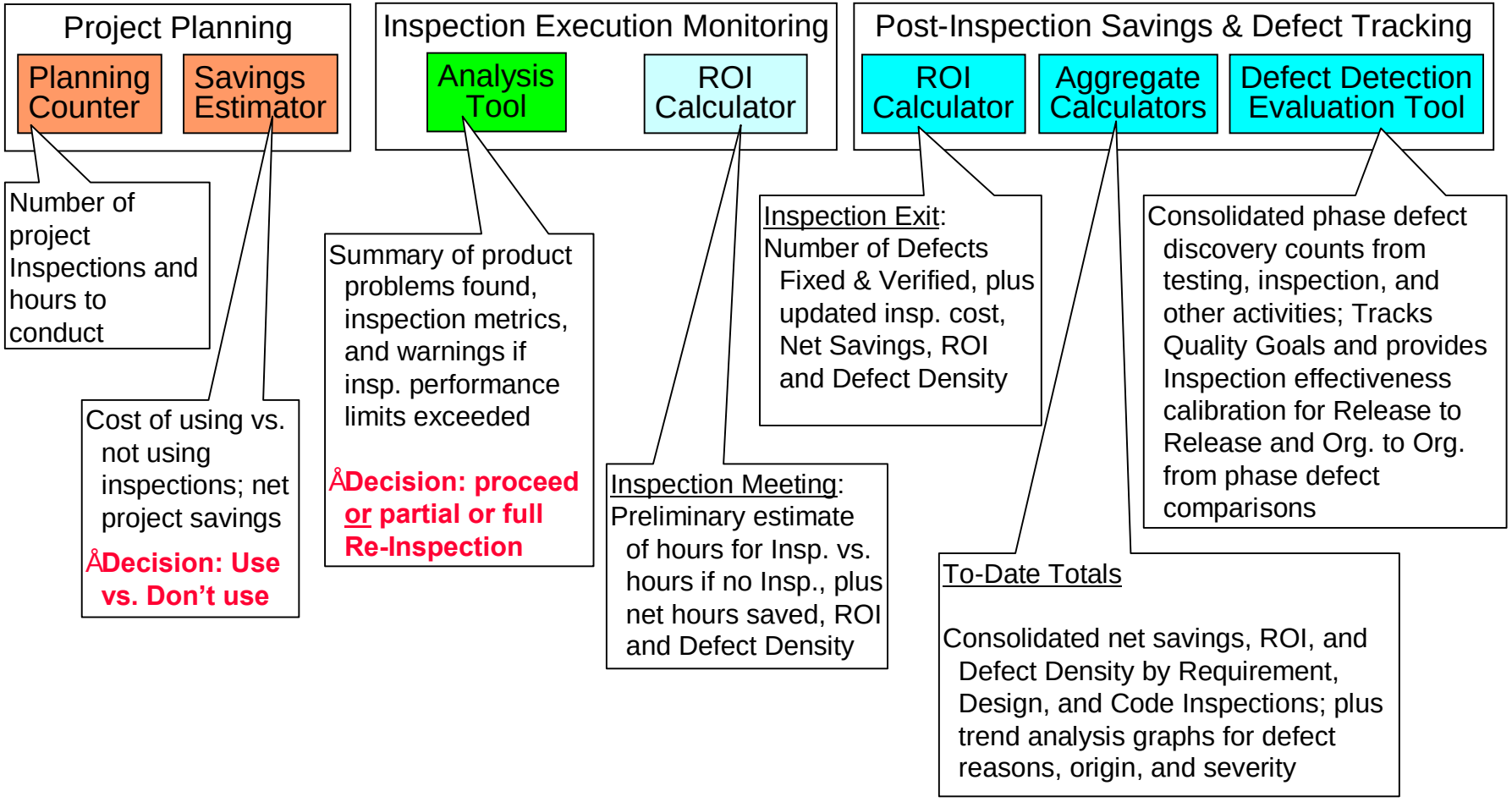



No company Inspection process capture

Pitfall Risks:

- No inspection process reference
  - Inspection training material not effective long-term reference
- Process misunderstood, inconsistent and incorrectly executed
- No repository for projectôs Inspection ólessons learnedô
- Inspections not identified and included in company culture and SDLC
  - Incorrectly viewed as stand-alone process
- Inspections fail

# Inspection Tools for Management



# Management Inspection Performance Class

Objective:

- Provide management with understanding of their responsibilities and the tools to be successful for the short and long term

Agenda:

1. What can inspections do for you
2. What is the inspection process
3. What is managementôs role/responsibilities
4. How to plan for inspections
5. How to estimate inspection savings
6. How to collect inspection data
7. How to analyze inspection data
8. How to deal with inspection pitfalls & challenges

# Inspection Class Agenda for Practitioners

- - - - - Half Day - - - - -

1. Teams and Case Study Material
   Course Objectives
2. Developing with Inspections
   and the Role of Process
3. Inspection Process Objectives
   Where and When to Inspect
4. Inspection Process
   Inspection Defect Definitions
5. Inspector Selection, Roles
   and Responsibilities
6. Inspection Scenario

   Team Case Study
   - Part1: Code or Non-Reqt. Doc.
   - Part2: Reqt. Doc
7. Evaluating Inspection Data
8. Inspection Situations
9. Inspection Savings
   Management Roles/Responsibilities
10. *Getting Started!*

- - - One Day - - -

Quarter Day

- - - Quarter Day - - -

Course consistent with IEEE Standard 1028 for Inspections
(IEEE - Institute of Electrical and Electronics Engineers, Inc.)

# Objectives of the Stewart–Priven Group (SPG), LLC

- Expand the industry use of best-practice quality software inspections by introducing an "Inspection Methodology" which provides for:
  1. A comprehensive set of management inspection tools for planning, projecting savings, monitoring, and tracking inspections across projects and organizations
  2. Ensuring client development infrastructure can support inspections
  3. Rapid inspector training including computerized tools, checklists, and reference card
  4. Follow-on support for management and inspectors
  5. Compliance with "IEEE Standard 1028 for Inspections"
- SPG was formed in 2006 by Roger Stewart and Lew Priven
  - Both have extensive development experience and experience in implementing, managing, and training the inspection process
  - Collectively they've trained over 5,000 inspectors at more than 50 company locations in the U.S., Europe and Japan for Michael Fagan Associates
- Roger and Lew co-authored "How to Avoid Software Inspection Failure and Achieve Ongoing Benefits," published in CROSSTALK Jan. 2008
- Roger and Lew spoke at the June 2007 SSTC and the May 2008 PSQT conferences about removing common pitfalls associated with implementing inspections, and use of effective computerized Mgmt. and inspector tools

# Lew Priven

Lew Priven is an experienced executive with an extensive management and technical background ï including system and software development, software quality training, management development training, human resource management, and executive management.

Prior to co-founding the Stewart-Priven Group, LLC, Lew was an Associate with Michael Fagan Associates, where he trained over 2,000 students at 24 company locations in the Fagan Inspection Process.

Before joining Michael Fagan Associates, Lew was Chief Operating Officer of Wellspring Resources, LLC. Among his responsibilities at Wellspring, he managed the rapid growth of the Washington, DC application development center. To help manage the growth, he introduced and made extensive use of inspections to insure the quality and timely delivery of software supporting the delivery of outsourced benefits administration services.

Lew was Vice-President of Engineering and Application Development at General Electric Information Services (GEIS). While at GEIS he introduced inspections for verifying the accuracy and usability of procedures for the introduction of software and network upgrades.

Lew held a number of positions at IBM including: Vice President of Application Development for IBMôs Application Systems Division, Director of Operations and Development for the IBM Information Network, and Vice President of Information Technology and Human Resources for Satellite Business Systems. He also served as a Director on the Corporate Technology Staff.

During development of IBMôs MVS operating system, Lew formed and managed the team, led by Michael Fagan, to improve the quality of software. The teamôs work resulted in the development of the inspection process.

In addition to his work on inspections, Lew received an IBM Outstanding Contribution Award for developing the Continuous Integration process.

Lew has a BS in Electrical Engineering from Tufts University and an MS in Management from Rensselaer Polytechnic Institute.

# Roger Stewart

Roger Stewart is an experienced Lead Systems Engineer and Program Manager with an extensive management and technical background ï  including Systems Engineering, Software Development, System Integration, System Testing, and Process Improvement.

Prior to co-founding the Stewart-Priven Group LLC, Roger was an Associate with Michael Fagan Associates, where he trained over 3,000 students at 30 company locations in the Fagan Inspection Process.

Before joining Michael Fagan, the founder of software inspections, Roger worked for IBM Federal Systems, and later Lockheed-Martin Federal Systems, developing large complex software systems for the Space Shuttle, Satellite Command and Control, Air Traffic Control, and military avionics systems. He also worked in the commercial sector developing Banking, Telecommunications, and Networking and Operating system software.

Roger's first use of formal inspections occurred while managing the On-Board Space Shuttle Guidance, Navigation and Computer Redundancy Management software. For the rest of his career, inspections formed an integral part of projects he managed and worked on. (The On-Board Space Shuttle software program was the first to earn the coveted SEI CMM Level 5 rating.)

Roger was the architect and lead author for advancing the Software Development Life Cycle for the IBM Telecommunications Systems Development Process, which included introducing and defining their inspection process. This work was recognized with an IBM Communications Division level award.

While on a 3-year assignment in Sydney Australia to manage the development of a new generation of banking applications and services, Roger was the Program Manager for defining and implementing the Software Development life cycle - including the introduction and definition of inspections.

While the Test Architect for Air Route Traffic Control System Upgrade programs, Roger defined the end-to-end test and acceptance life cycle and the inspection process used.

Roger has a BS in Mathematics from Cortland University.