CHICAGO QUALITY ASSURANCE ASSOCIATION
Member of the QAI Global Community
CQAA

pearl street group

Larry Dribin, Ph.D.

Pearl Street Group, Inc.

larry@pearlstreetgroup.com

Cell: 847-807-7390

Metrics for Testing Excellence!

V1-0

November 18, 2009

Larry’s Background

- Process Improvement & Metrics Consultant
- Adjunct Professor, Software Engineering, DePaul University

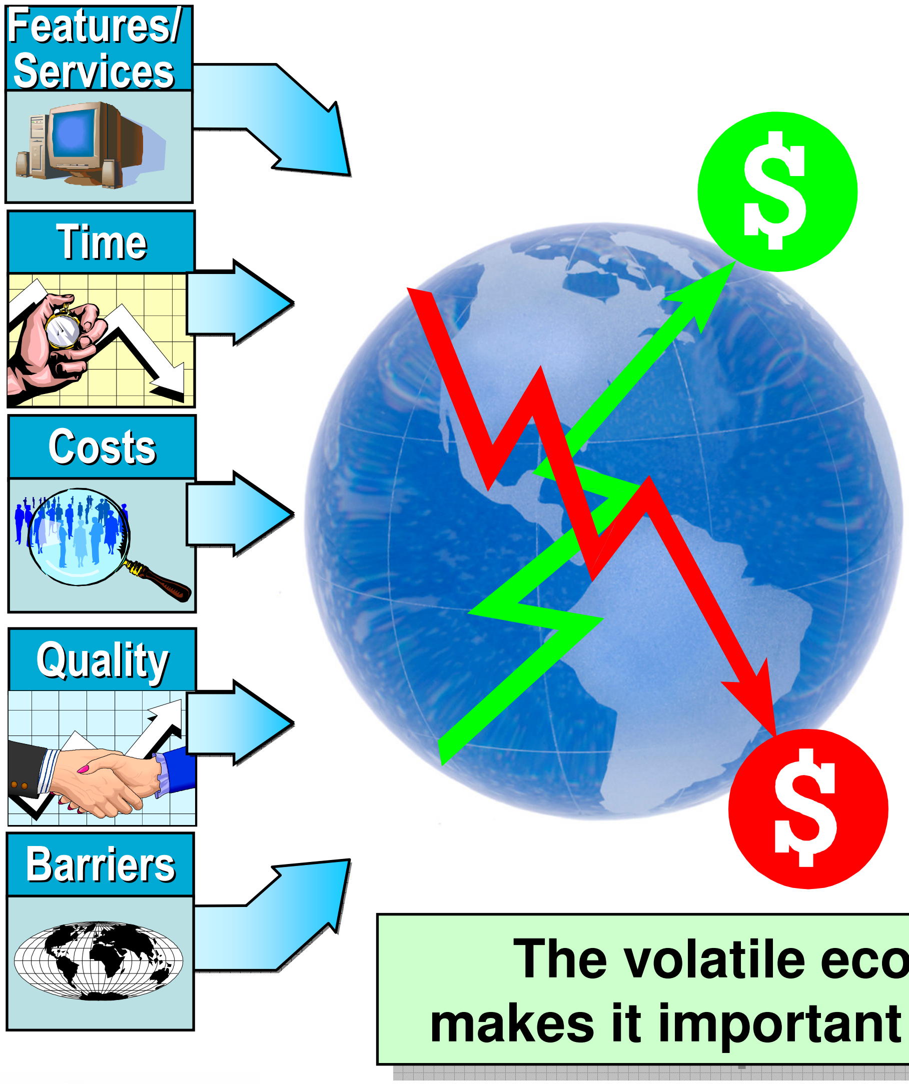
PSG Services	Partial Client List	
Business & IT Process & Measurement Improvement Services: ♦ Process Development ♦ Measurement Development Project Management Services: ♦ Critical Project Recovery ♦ PMO Implementation ♦ Portfolio Alignment	♦ ABN Amro ♦ Advocate Medical Group ♦ Alltel Information Systems ♦ Allstate ♦ American Electric Power ♦ American Red Cross ♦ Assurant Health ♦ AT & T ♦ Baan ♦ Bear Stearns	♦ Eli Lilly ♦ GEICO ♦ Great West Healthcare ♦ Highmark Blue Cross ♦ JP Morgan Chase ♦ Procter & Gamble ♦ QuadTech ♦ Teleflex Medical ♦ Tribune Corporation ♦ United Plastics Group ♦ US Bank ♦ Walgreens ♦ Western Digital

Agenda

- ♦ **Testing Process Improvement**
- ♦ Linking Process Improvement and Measurement
- ♦ Testing Process and Metrics
- ♦ Testing Measurement Program Design
- ♦ Summary

The goal for this workshop is to discuss how measurement and process improvement can be used to improve a testing organizations performance.

Our World – Volatile Global Competition



To survive, organizations must constantly improve by:

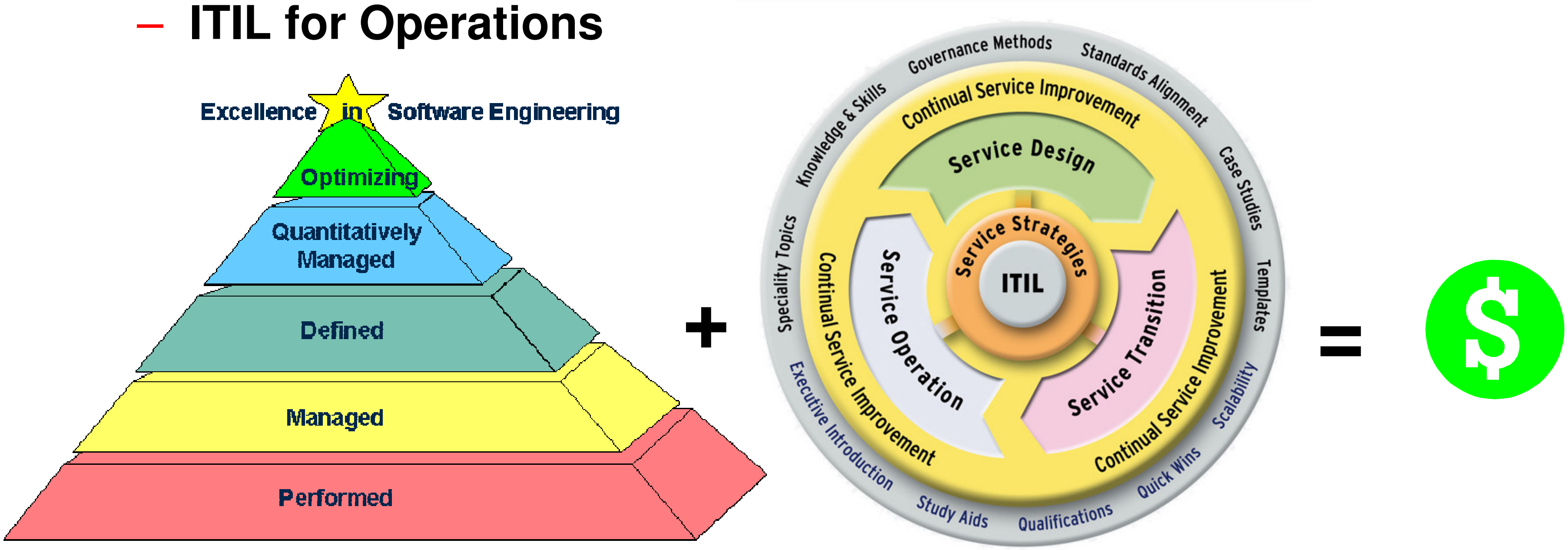
- Creating new products & services
- Improving existing products and services
- Improving quality
- Reducing cycle time and costs
- Innovation

The volatile economic environment makes it important to constantly improve!

IT is Constantly Striving to Improve

IT Organizations use Best Practice Process Improvement Frameworks to improve:

- Agile and/or CMMI for Development
- ITIL for Operations



CMMI or Agile

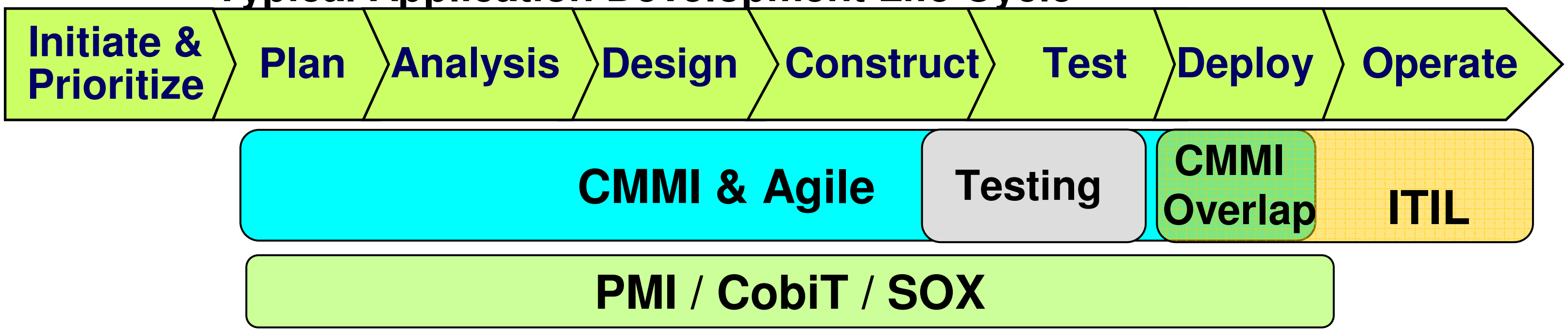
ITIL

What about Testing?
Agile provides some focus on testing, but neither CMMI or ITIL focus on Testing!

What is a “Best Practices” Frameworks?

- “Best Practices” in general are statements of common sense
- Frameworks overlap in some areas
- Some areas are not covered by major frameworks
- May use different language to describe the same processes

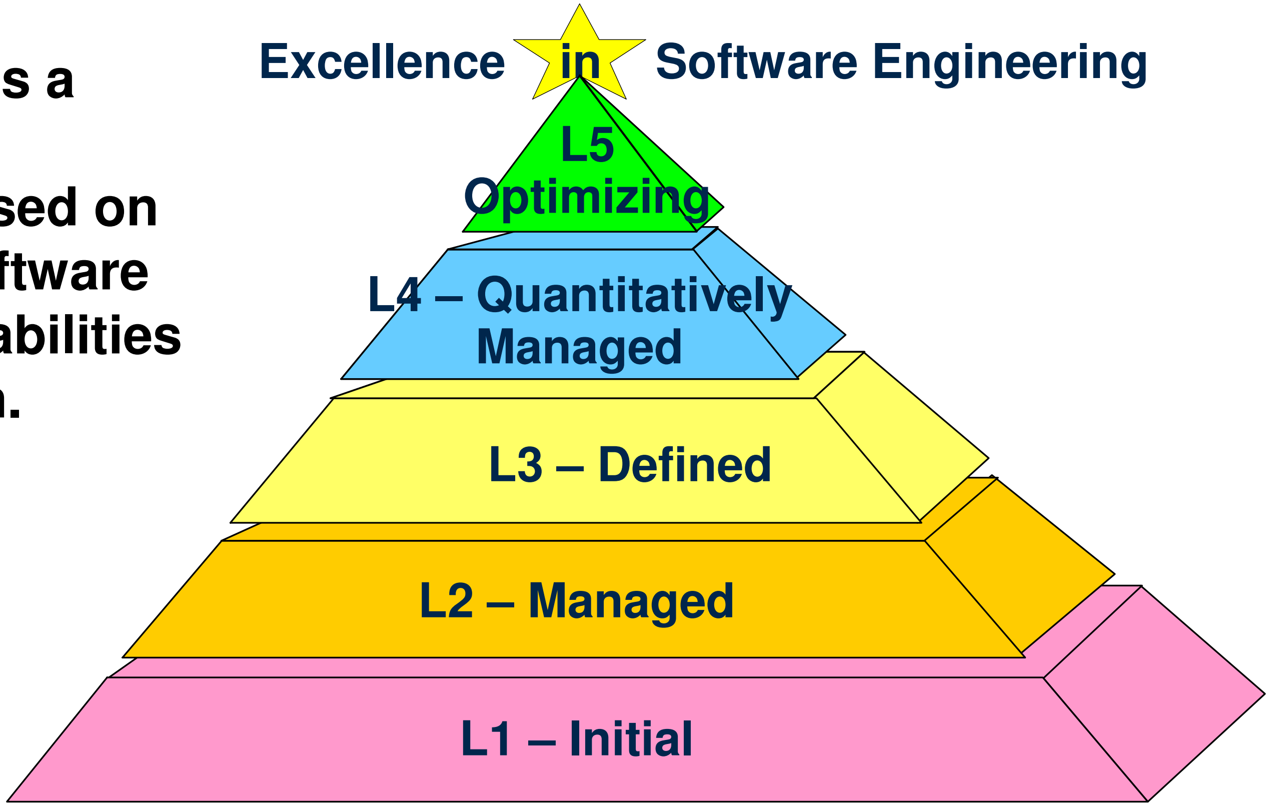
Typical Application Development Life Cycle



CMMI	– Capability Maturity Model Integrated
CoBit	– Control Objects for Information Technology
ITIL	– IT Infrastructure Library
PMI	– Project Management Institute (PMBOK)
SOX	– Sarbanes-Oxley
Testing	– CSTE CBOK (not a full Process Framework)

Capability Maturity Model Integrated (CMMI)

- The CMMI model is a “Best Practice Framework” focused on improving the software development capabilities of an organization.
- Supports
  - Waterfall
  - Agile


Excellence in Software Engineering
L5 Optimizing
L4 – Quantitatively Managed
L3 – Defined
L2 – Managed
L1 – Initial

The CMMI Model uses a five level process maturity rating system similar to a five start hotel rating system to assess an IT organizations capability to develop software.

CMMI for Development v1.2 Process Areas

Level 1 – Initial

- Ad Hoc

Level 2 – Managed

- Requirements Management
- Project Planning
- Project Monitoring & Control
- Supplier Agreement Management
- Measurement & Analysis
- Process and Product Quality Assurance
- Configuration Management

Level 3 – Repeatable

- Requirements Definition
- Technical Solutions
- Product Integration
- Verification
- Validation
- Organizational Process Focus
- Organizational Process Definition + IPPD
- Organizational Training
- Integrated Project Management + IPPD
- Risk Management
- Decision Analysis & Resolution

Level 4 - Quantitatively Managed

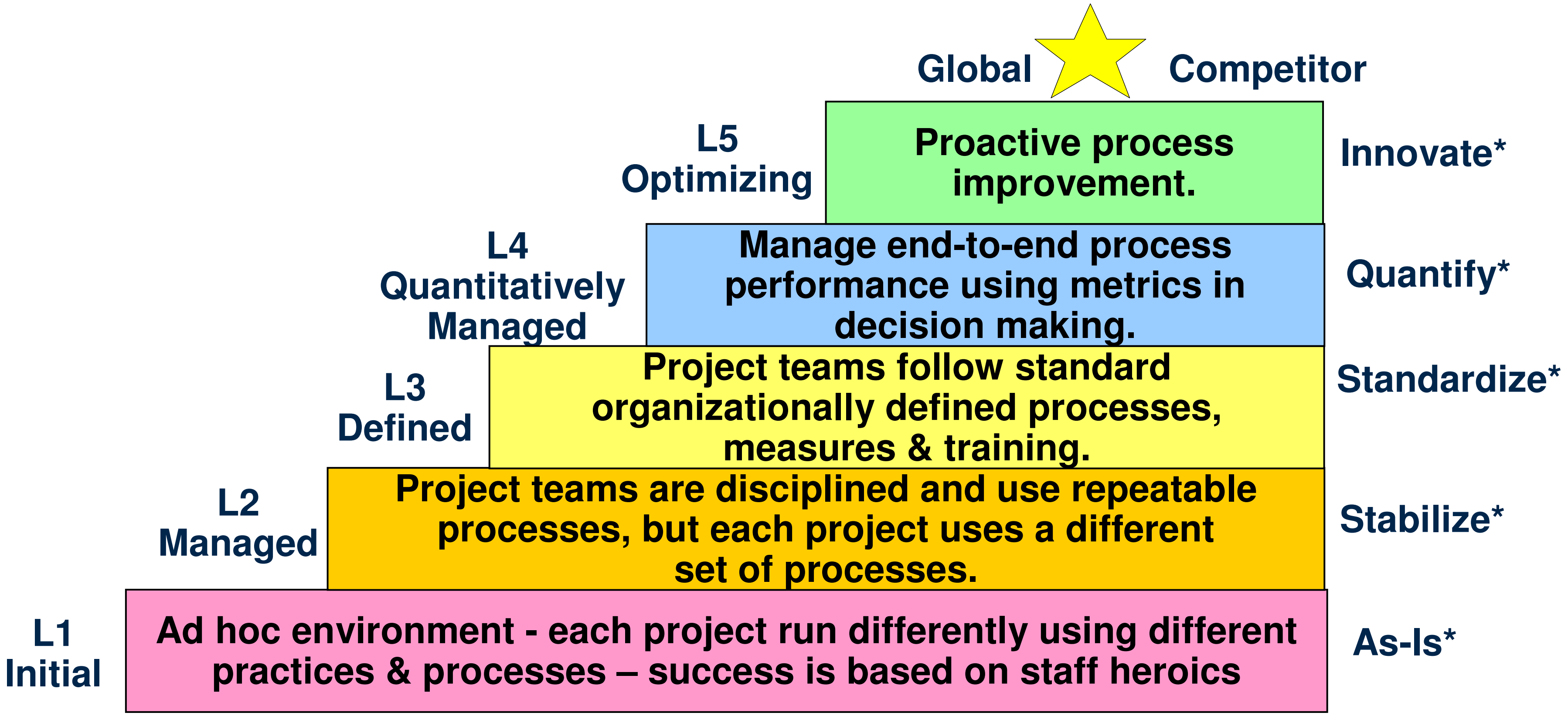
- Organizational Process Performance
- Quantitative Project Management

Level 5 - Optimizing

- Organizational Innovation and Deployment
- Causal Analysis & Resolution

CMMI says little about Testing – focus is on Peer Reviews.

The Heart of the CMMI – Process Maturity



Global Competitor

L5 Optimizing – Proactive process improvement. – Innovate*

L4 Quantitatively Managed – Manage end-to-end process performance using metrics in decision making. – Quantify*

L3 Defined – Project teams follow standard organizationally defined processes, measures & training. – Standardize*

L2 Managed – Project teams are disciplined and use repeatable processes, but each project uses a different set of processes. – Stabilize*

L1 Initial – Ad hoc environment - each project run differently using different practices & processes – success is based on staff heroics – As-Is*

* Source: Bill Curtis, ITMPI Presentation, Nov. 2008.

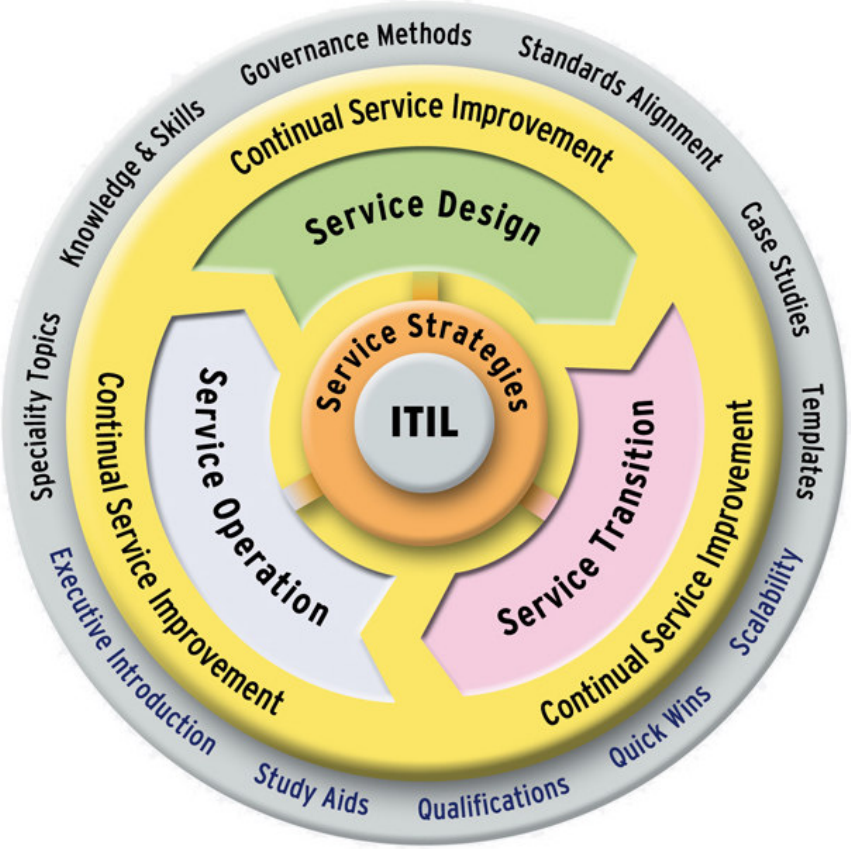
CMMI organizational assessment & process maturity concepts can be applied to other Best Practice Frameworks with great success.

ITIL (IT Infrastructure Library)

ITIL is a framework of "Best" practices developed to provide high quality IT operations

- Version 3 – Released in June, 2007
- Unlike the CMMI, ITIL focus is on individual certification versus organizational improvement
- ITIL Processes in v3.0 are not as rigorously defined as the process areas in CMMI

Since its inception, ITIL has won wide acceptance in IT Operations



ITIL v3.0 Process Areas

Service Strategy

- Strategic Assessment
- Service Portfolios
- Return on Investment
- Financial Management
- Organizational Development

Service Design

- Service Catalogue Management
- Service Level Management
- Capacity Management
- Availability Management
- IT Service Continuity Management
- Information Security Management
- Supplier Management

Service Transition

- Asset and Configuration Management
- Release and Deployment Management
- Service Validation and Testing Releases

Service Operation

- Event Management
- Incident Management
- Request Fulfillment
- Problem Management
- Access Management
- IT Operations Management
- Application Management
- Service Operation and Project Management

Continuous Service Improvement

ITIL talks about service validation, but says little about how to do this.

Testing Process Frameworks

- TMM – Testing Maturity Model (I. Burnstein – IIT)
  - Multi-Level Model, but not well integrated into CMMI Architecture
  - Recent new focus by TMMI Foundation
- TPI – TMap – Test Process Improvement and Test Management Approach (T. Koomen, M. Pol, Sogeti)
  - Robust and well documented
  - Multi-Level Model, but not well integrated into CMMI Architecture
- TPAM – Test Process Assessment Model (Y. Chernak, Valley Forge Consulting)
  - Not well documented, but provides the best integration with the CMMI through Key Practices
    1. Test Strategy Planning
    2. Test ware Design and Maintenance
    3. Test Execution and Reporting

Individual certifications like the CSTE are not full organizational process frameworks. None of these frameworks has won the acceptance of the CMMI or ITIL.

Suggestion – Tailor your Own

- "Best Practice" Process Frameworks

  Provide guidelines, but do not provide enough details to develop deployable processes

- Commercial Process Models such as RUP or Microsoft Solution Framework need to be tailored
- So, develop your own Process Model
  - Based on "Best Practices"
  - CMMI & Agile for Development
  - ITIL for Operations
  - TPAM, TMAP – TPI and TMM for testing
- Define enough detail so that processes provide useful guidance and can be deployed

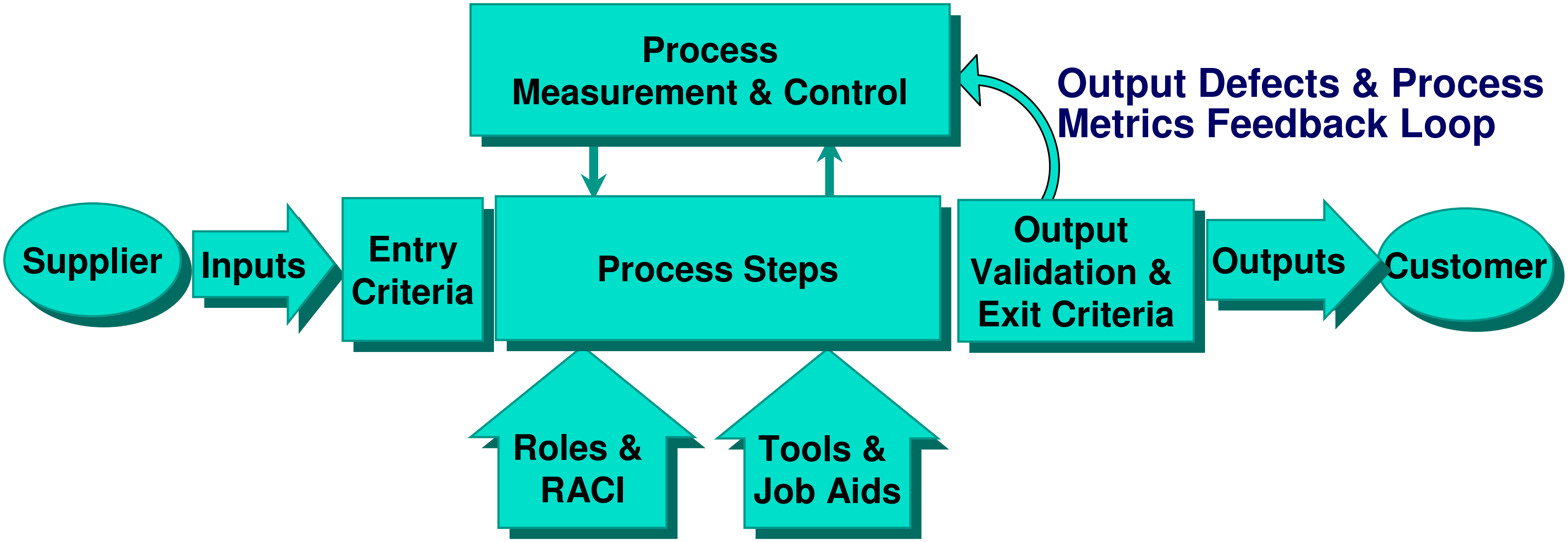
Use the "Best Practice" frameworks and "Body's of Knowledge" as a guide to developing your own detailed, deployable process models.

Agenda

- Testing Process Improvement
- **Linking Process Improvement and Measurement**
- Testing Process and Metrics
- Testing Measurement Program Design
- Summary

Processes and Measurement

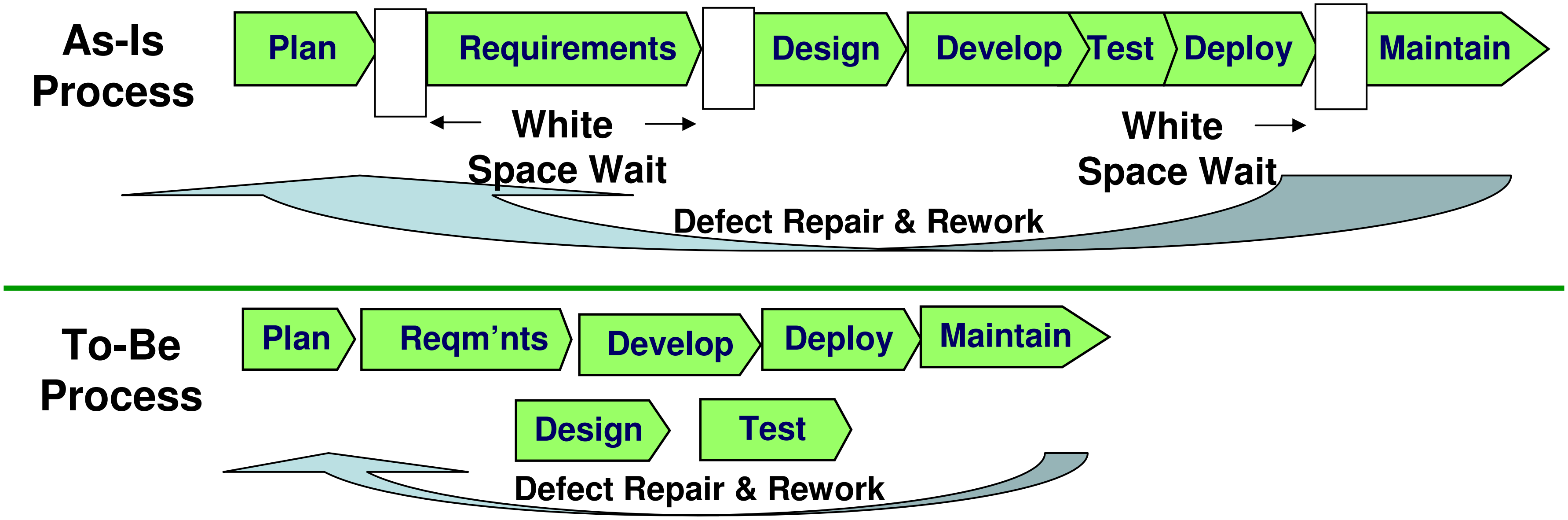
- ♦ Processes are organized sequences of activities and practices that transform inputs into outputs
- ♦ Processes are the foundation for the organization's work
- ♦ Processes can be improved once they become repeatable
- ♦ Measurement is essential for process improvement


Process Measurement & Control
Output Defects & Process Metrics Feedback Loop
Supplier
Inputs
Entry Criteria
Process Steps
Output Validation & Exit Criteria
Outputs
Customer
Roles & RACI
Tools & Job Aids

"Best Practice" frameworks require repeatable processes and measurement!

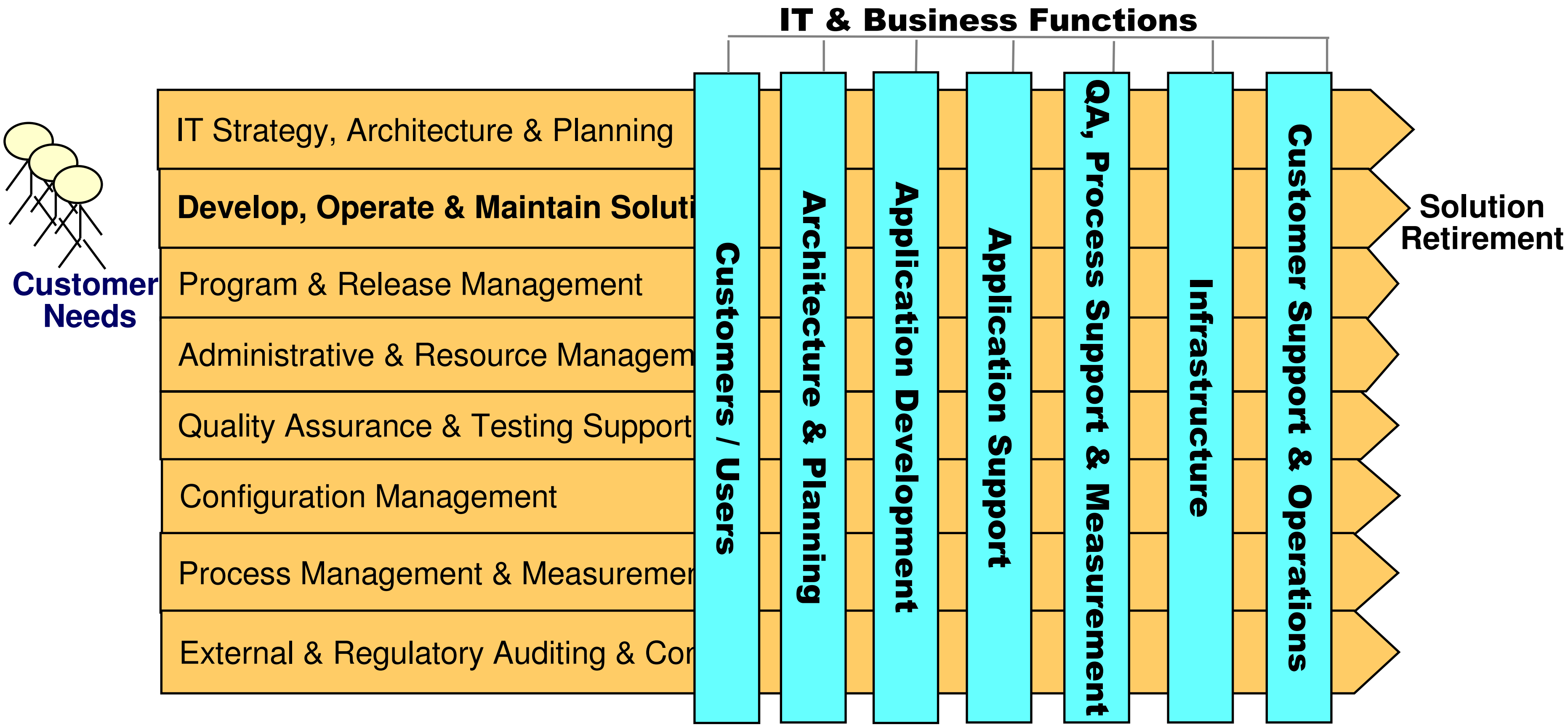
How Process Improvement Works

1. Eliminate White Space Wait Time between processes
2. Reduce Process Execution Time for Requirements
3. Overlap the Develop Process with Design & Testing Processes
4. Reduce defects & rework with better QA Reviews



Result: Less rework, shorter cycle time, higher productivity, lower costs and a foundation for innovation!

Core Processes Cross Organizational Functions



IT & Business Functions

Customer Needs

IT Strategy, Architecture & Planning

Develop, Operate & Maintain Soluti

Program & Release Management

Administrative & Resource Managem

Quality Assurance & Testing Support

Configuration Management

Process Management & Measuremer

External & Regulatory Auditing & Cor

Customers / Users

Architecture & Planning

Application Development

Application Support

QA, Process Support & Measurement

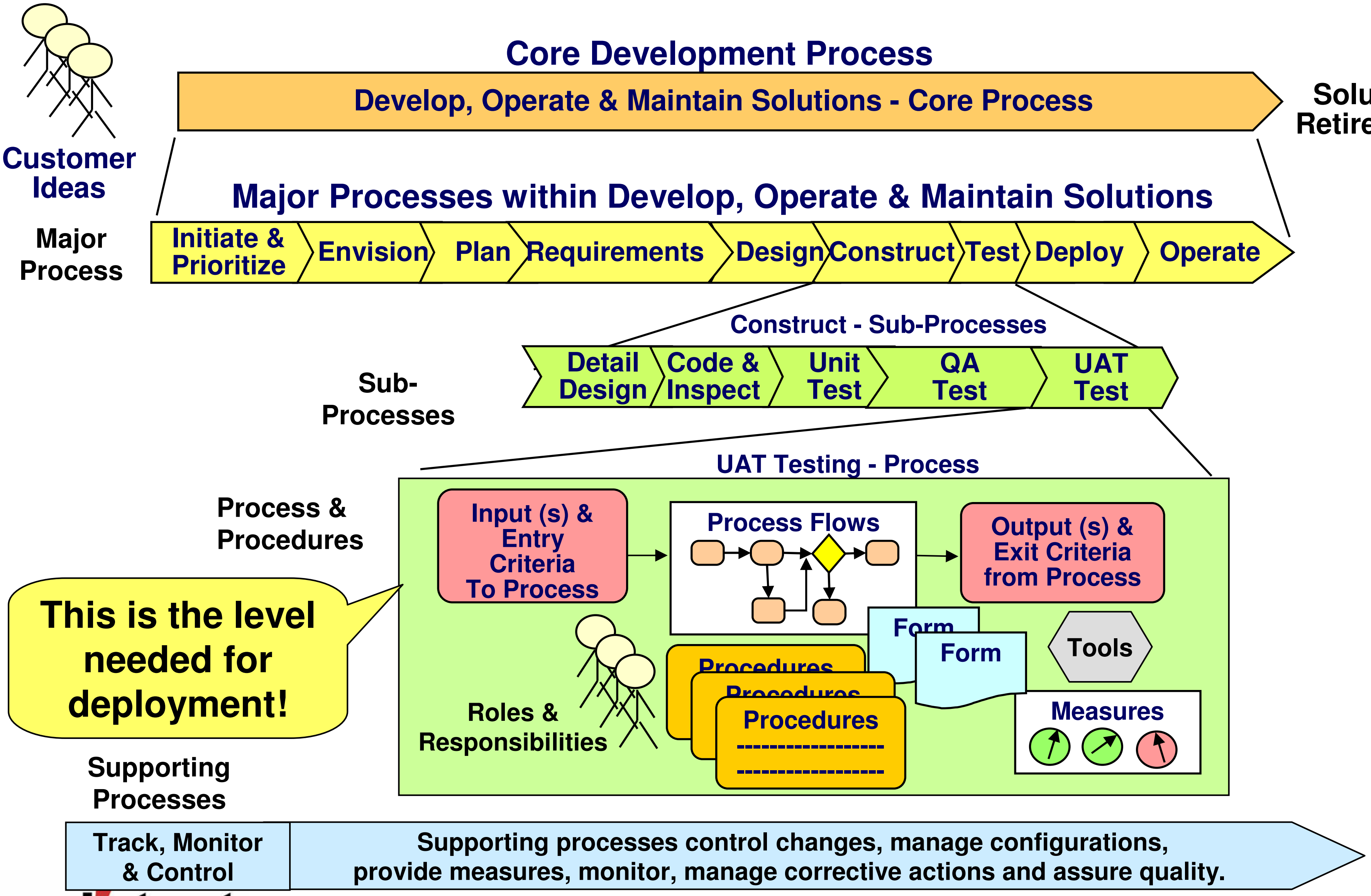
Infrastructure

Customer Support & Operations

Solution Retirement

Functional organizations need to support the core processes by implementing them in their functional areas.

Process Decomposition



Customer Ideas

Core Development Process

Develop, Operate & Maintain Solutions - Core Process

Solution Retirement

Major Processes within Develop, Operate & Maintain Solutions

Major Process

Initiate & Prioritize | Envision | Plan | Requirements | Design | Construct | Test | Deploy | Operate

Construct - Sub-Processes

Sub-Processes

Detail Design | Code & Inspect | Unit Test | QA Test | UAT Test

UAT Testing - Process

Process & Procedures

Input (s) & Entry Criteria To Process

Process Flows

Output (s) & Exit Criteria from Process

Form

Form

Tools

Procedures

Procedures

Procedures

Measures

Roles & Responsibilities

This is the level needed for deployment!

Supporting Processes

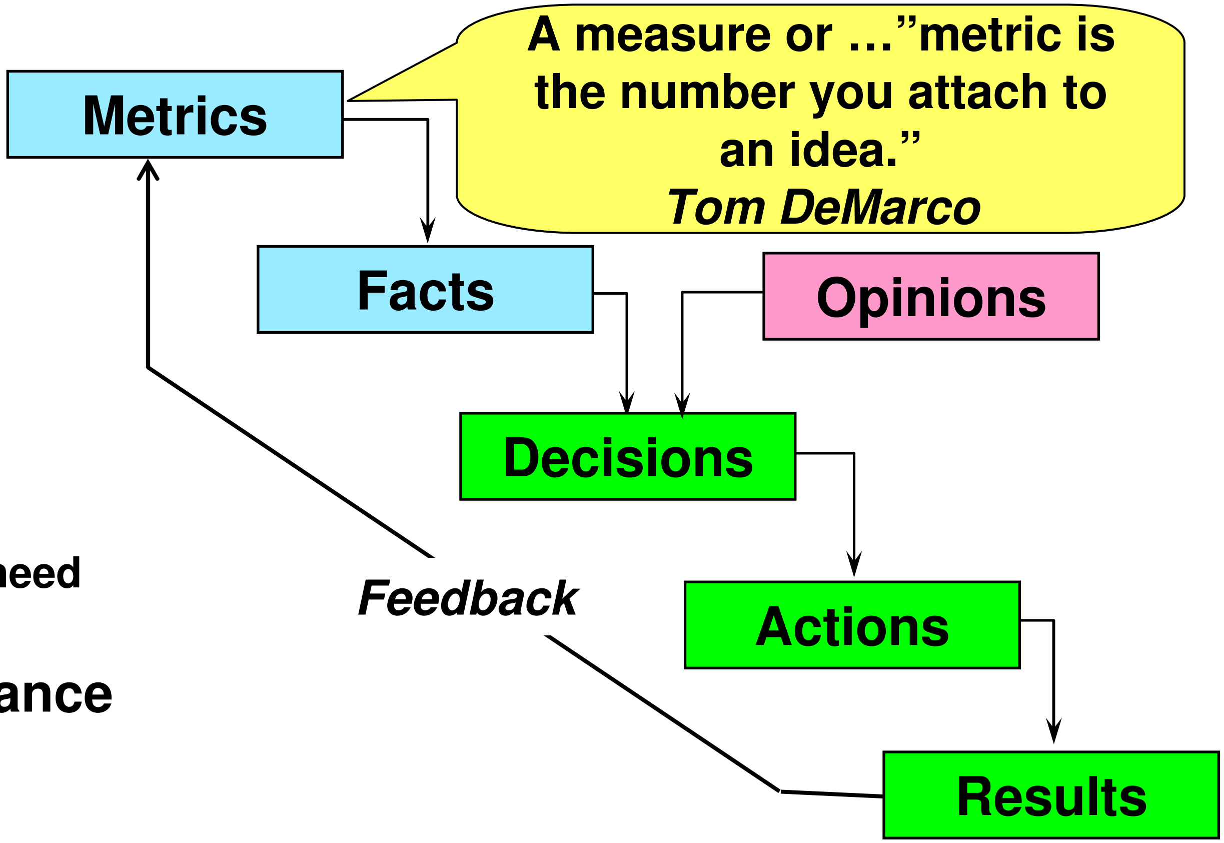
Track, Monitor & Control

Supporting processes control changes, manage configurations, provide measures, monitor, manage corrective actions and assure quality.

What Get’s Measured, Get’s Done!

We measure:

- To Communicate
  - With Management
  - With Peers
  - With Subordinates
- To Manage
  - Make Decisions
  - Estimate
    - How much we will need
    - What will it cost
- To Improve Performance
  - Quality
  - Cycle Time
  - Productivity
  - Risk



A measure or …”metric is the number you attach to an idea.”
Tom DeMarco

What Doesn’t Get Measured, Doesn’t Get Done!

Measurement is Hard to Do

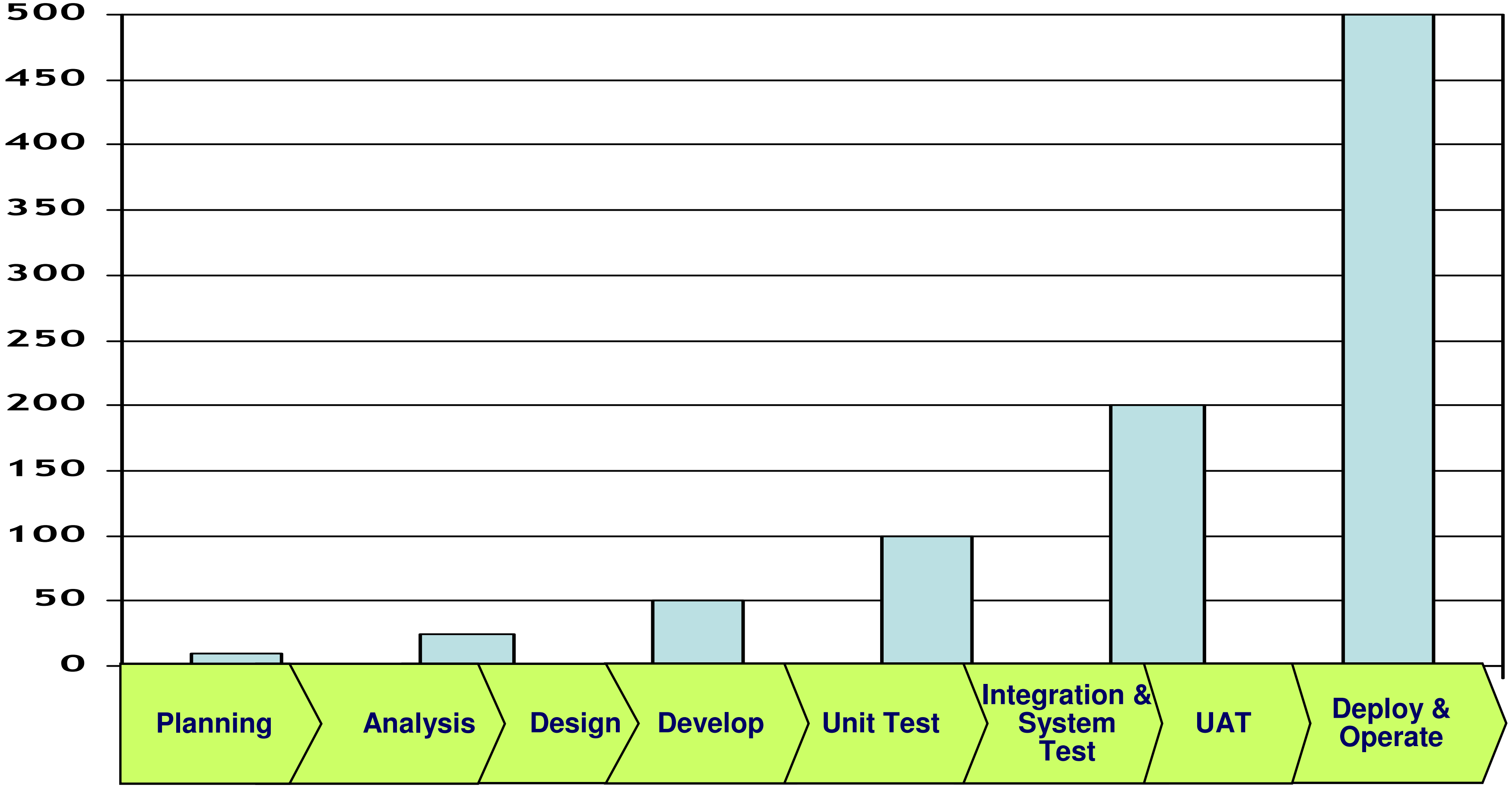
- Difficult to design good measures
- Inadequate measurement infrastructure
- Lack of adequate measurement expertise
  - Good measurement takes time and expertise
- Staff often does not want to be measured and has a a Love – Hate Relationship with Measurement
  - Love it for other folks – i.e.. Sports Statistics
  - Love it when are doing well – Low golf handicap
  - Hate it when we are not doing well – Forget golf handicap

The root cause of these issues is often due to inadequate measurement resources, knowledge and expertise in the organization!

Agenda

- Testing Process Improvement
- Linking Process Improvement and Measurement
- Testing Process and Metrics
- Testing Measurement Program Design
- Summary

Defects are Expensive



Rework costs rise dramatically when defects are found late in the SDLC!
Rework cost depend on where defect was Inserted and where it was found.

Testing Terms

Mistake (Bug)

A error or flaw in requirements, design, or coding that can cause an information system from behaving as intended. Mistakes occur ***before*** the application has been inspection, tested and gone into production. People are human and make mistakes . Thus some small number of mistakes are inevitable. Testing is a key way of preventing mistakes from becoming defects.

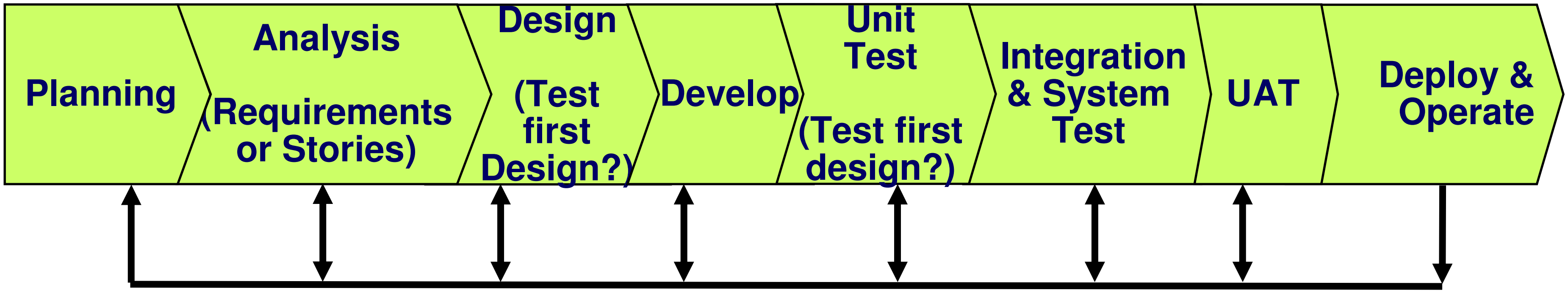
Defect

A error or flaw in requirements, design or coding that causes an information system from behaving as intended ***after*** it has been tested and is gone in production. Defects result when a mistake is allowed to reach a customer and can be prevented by through inspection and testing.

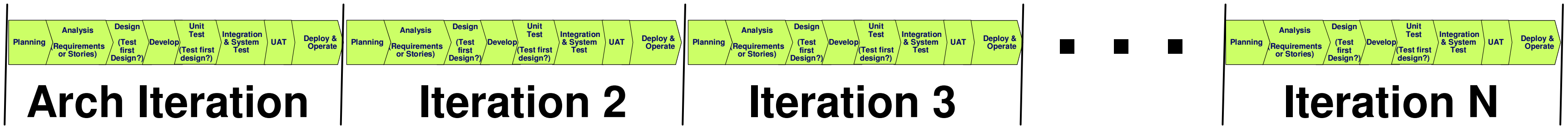
Our job as testers is to prevent Mistakes from becoming Defects!

Software Development Life Cycles

- Software development follows an set of steps with feedback from each step to one of the prior steps if defects, inconsistencies, or missed requirements are found
- In practice, there are two main variations of these general steps:
  - Waterfall



  - Agile

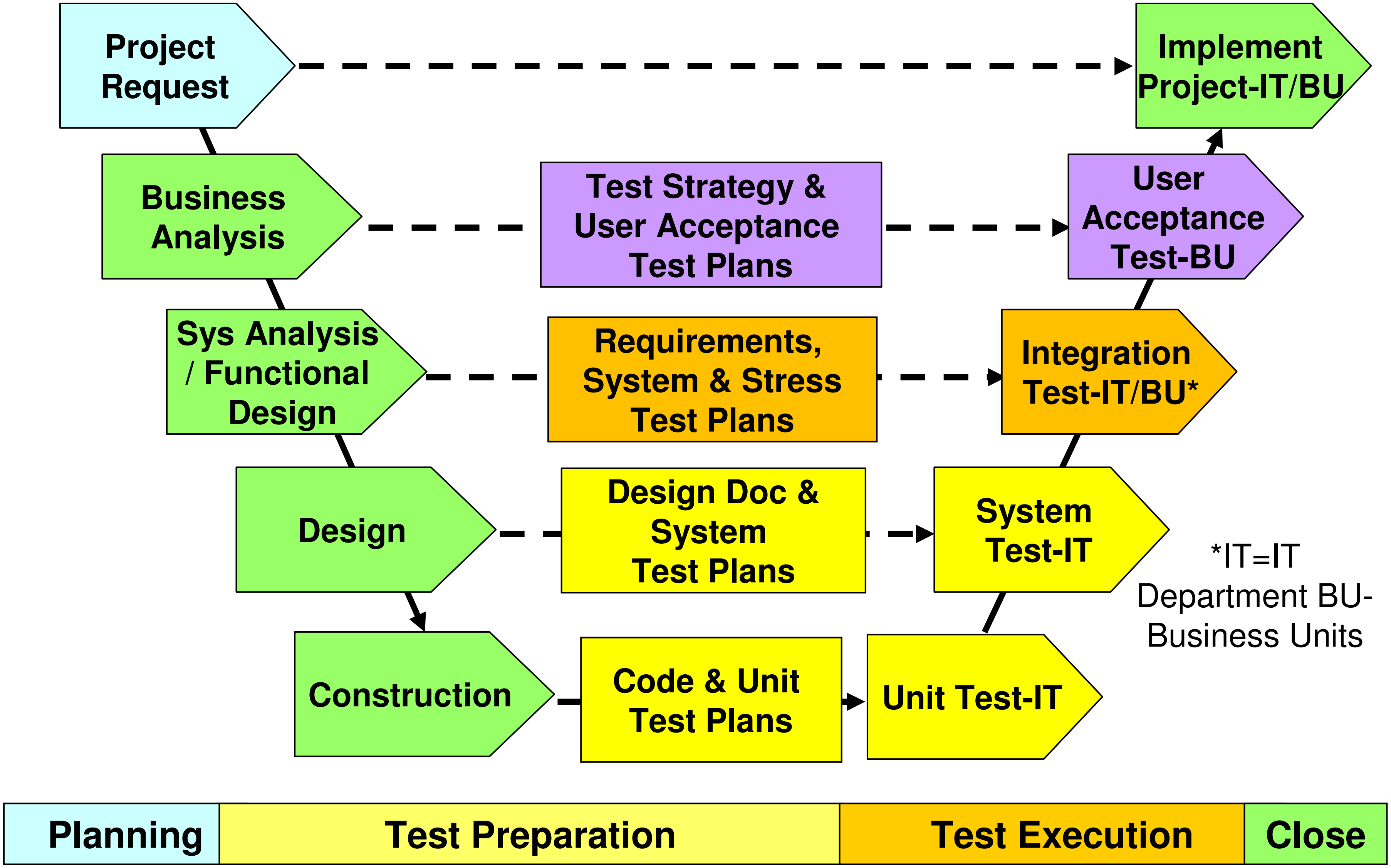


Similar steps occur for a Waterfall project or within an Agile Iteration

Structured Testing

- Structured testing is a process for organizing the testing activities in a fashion that can lead to continuous improvement
- Structured testing mirrors the Software Development Life Cycle and is made up:
  - Test Planning
  - Test Preparation – Test Case Development / Specification
  - Test Execution
  - Test Completion
- Structured testing covers all testing activities
  - Unit
  - Integration & System (QA)
  - Performance and stress
  - User Acceptance Testing

V-Model of Testing with key Test Documents



*IT=IT Department BU-Business Units

Test Planning – Rule of Thumb ~ 20%

- Test Planning begins by understanding project goals
- Review all available documents and participate in as many planning and requirements meetings as possible
- Typically 20% to 30% of key information is not documented and is coved in people to people interactions
- Develop Test Strategy and begin Test Plan
  - Determine Test Objectives
  - Identify the areas / functions to be tested
  - Identify needed resources
  - Develop a strategy for testing:
    - The functions
    - The information system as a whole
  - Determine who will test what
- During this phase an early estimate of the testing effort and resources required to complete testing per the Test Strategy should be developed

Test Planning Measures focus on Estimating

- Similar to Development Project Estimating:
  - Effort / Cost
  - Duration
  - Size
  - Quality
  - Business and Technical Risk
  - Sounds easy, but …
- Example Estimating Measures
  - Number of Test Cases and Scripts to be developed
  - Effort and Duration to develop the test cases
  - Effort and Duration to execute a typical test case
  - Estimate of potential number of Defects to be found (Software Reliability Theory and historical data)

Test Planning often based on a % of the development effort.

Why not develop our own testing estimates!

Testing Estimating – Similar to Development

- Estimation Techniques:
  - Expert Judgment
  - Historical Data – Past Projects
  - Empirical and Theoretical Models (Formulas)
    - Number of Requirements or Story Points
    - Function Point
    - Test Points
  - Size vs. Effort based Estimation
    - Often need to do both
    - Size is better for process improvement
- Estimation Rules of Thumb:
  - Have people responsible for the work estimate the work
  - Obtain multiple estimates when possible for critical techniques
  - A good estimating technique is to develop Optimistic, Most Likely and Pessimistic estimates
  - Never force an estimate on project team

Simple Test Estimation Example

Project	Requirements	Scenarios	Scripts
P1859	16	22	122
P1753	3	5	21
P1997A	9	28	42
P2185	10	15	26
P1669	2	12	144
P2286	1	2	28
P1684	6	20	67
P1064	2	6	79
P46	5	11	105
P2341	9	16	70
P1029	10	35	83
Totals	73	172	787

Estimated Scenarios	Estimated Scenarios - Actual	% Error Script Scenario	Estimated Scripts	Estimated Scripts - Actual	% Error Script Estimate
38	16	13%	172	50	41%
7	2	10%	32	11	54%
21	-7	-16%	97	55	131%
24	9	33%	108	82	315%
5	-7	-5%	22	-122	-85%
2	0	1%	11	-17	-61%
14	-6	-9%	65	-2	-3%
5	-1	-2%	22	-57	-73%
12	1	1%	54	-51	-49%
21	5	7%	97	27	39%
24	-11	-14%	108	25	30%
172	0	0%	787	0	0%

Scenarios per Requirement	**2.4**
Scripts per Requirement	**10.8**

Need to calibrate the model over time

- **Our estimate would be 2.4 test scenarios for each requirement (requires a requirements document)**
- **Need to calibrate the model with historical data**

The price for not calibrating is erroneous or misleading estimates!

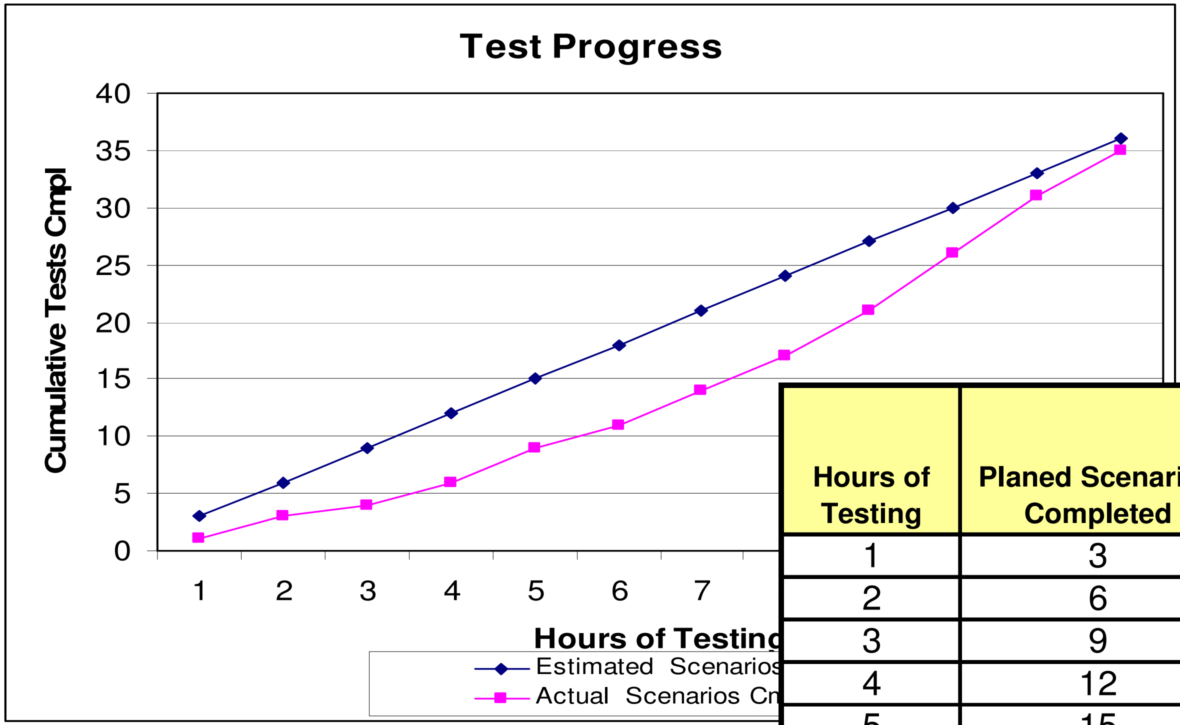
Test Preparation – Rule of Thumb ~ 40%

- During this phase the test case and related documents are developed and documented:
  - Use Cases
  - Test Scenarios, Test Cases and Test Scripts
  - Test Data and Test Data Bases
  - A special test environment may be required
- The Test Team participates in key meetings and review key documents to understand the business requirements and information system in general
- When completed, all key Test Documents should be validated ideally with an Inspection by key stakeholders

Test Preparation Measures focus on Progress

- Basic Project Management Monitoring and Control
- Focus on test plan and test case development
- Must align with the development schedule
- Example Measures:
  - Status of the Test Plan
  - Number of Test Cases developed
  - Test Case Churn (related to Requirements Churn)
  - Test Estimating Accuracy
  - Application Complexity
  - Test Coverage - Static
  - Use of Risk Based Testing
  - Defect Containment
  - Estimated of Number of Defects

Use Estimates and Actuals to Measure Progress



Hours of Testing	Planed Scenarios Completed	Actual Scenarios Completed	Actual Scenarios Completed per Hour
1	3	1	1
2	6	3	2
3	9	4	1
4	12	6	2
5	15	9	3
6	18	11	2
7	21	14	3
8	24	17	3
9	27	21	4
10	30	26	5
11	33	31	5
12	36	35	4

Estimate 3 scenarios per hour

Risk Based Testing Analysis

Risk = Likelihood * Impact

High Likelihood	Potential Partial Testing	Thorough Testing
Low Likelihood	Potential Minimal Testing	Potential Partial Testing
	Low Impact	High Impact

Even though the likelihood of a risk event is small, risk events can and often do happen.

Test Execution – Rule of Thumb ~ 40%

- Execute the Test Scenarios and Test Scripts and compare actual results to expected results
- Track Test Progress, Defects, monitor fixes and retest
- Check results of earlier testing to ensure that it was completed per the Test Plan – if not, log a Process Deviation
- Constantly check the test results to see if any aspects of the testing were overlooked or came to light based on the test execution
- Revise the Test Scenarios, Test Cases and Test Scripts as needed and as defects are identified

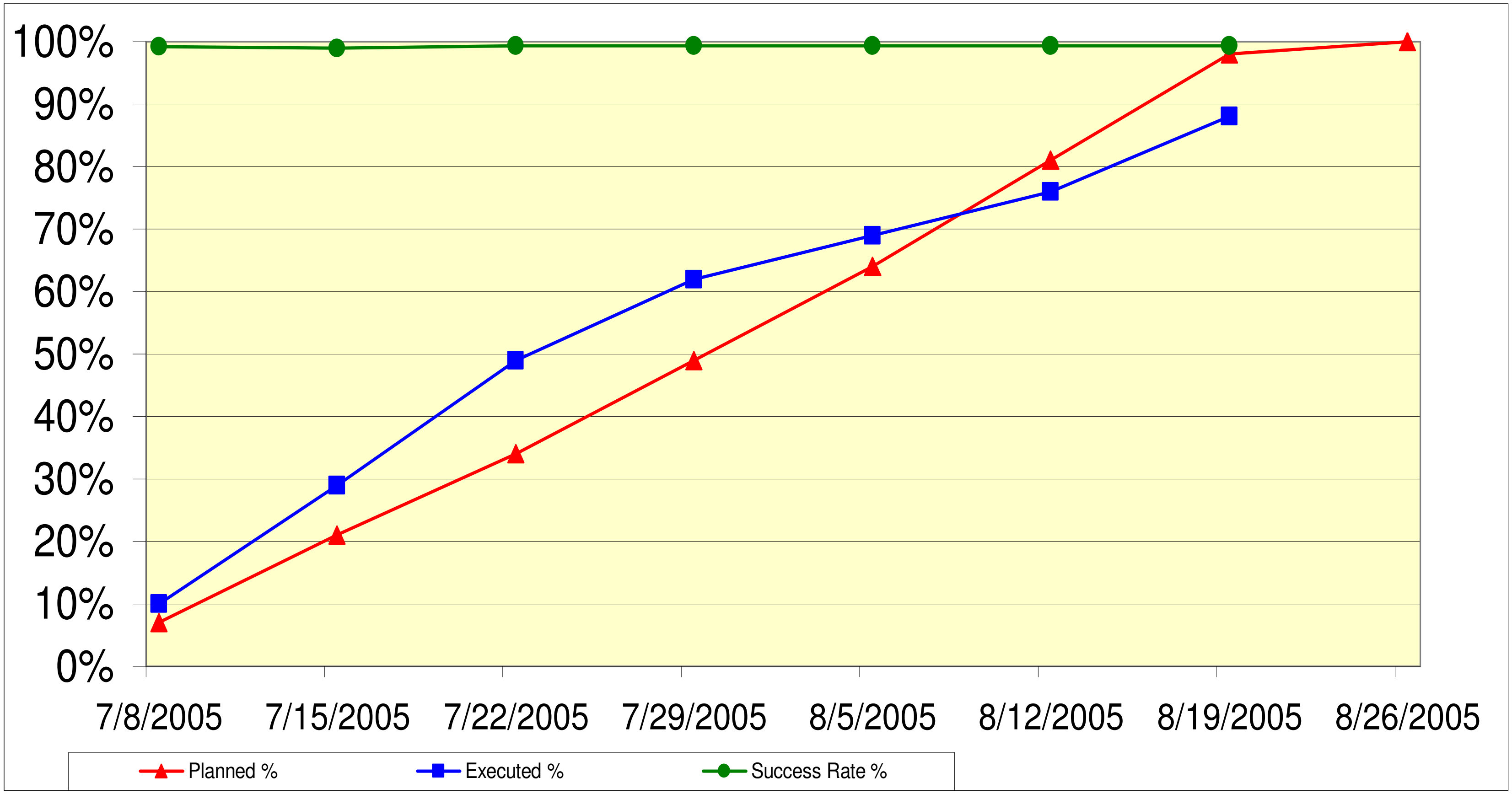
Approximately 15% of defects found in test are in the test cases or test data.

Test Execution Measures focus Test Execution and Defect Tracking

- As tests are executed, defects are found, repaired and retested
- Example Measures:
  - Test Cases Executed vs. Planned
  - Test Case Success Rate
  - Defect Run Rate
    - New Defects Discovered
    - Defects Closed
    - Defects Open
  - Defect Classification (Including bad fixes)
    - Category
    - Priority
  - Time to Fix a Defect
  - Test Coverage – Dynamic (Bill Curtis, CAST – Sept. 2009)
  - Application Complexity
  - Risk Based Testing Results

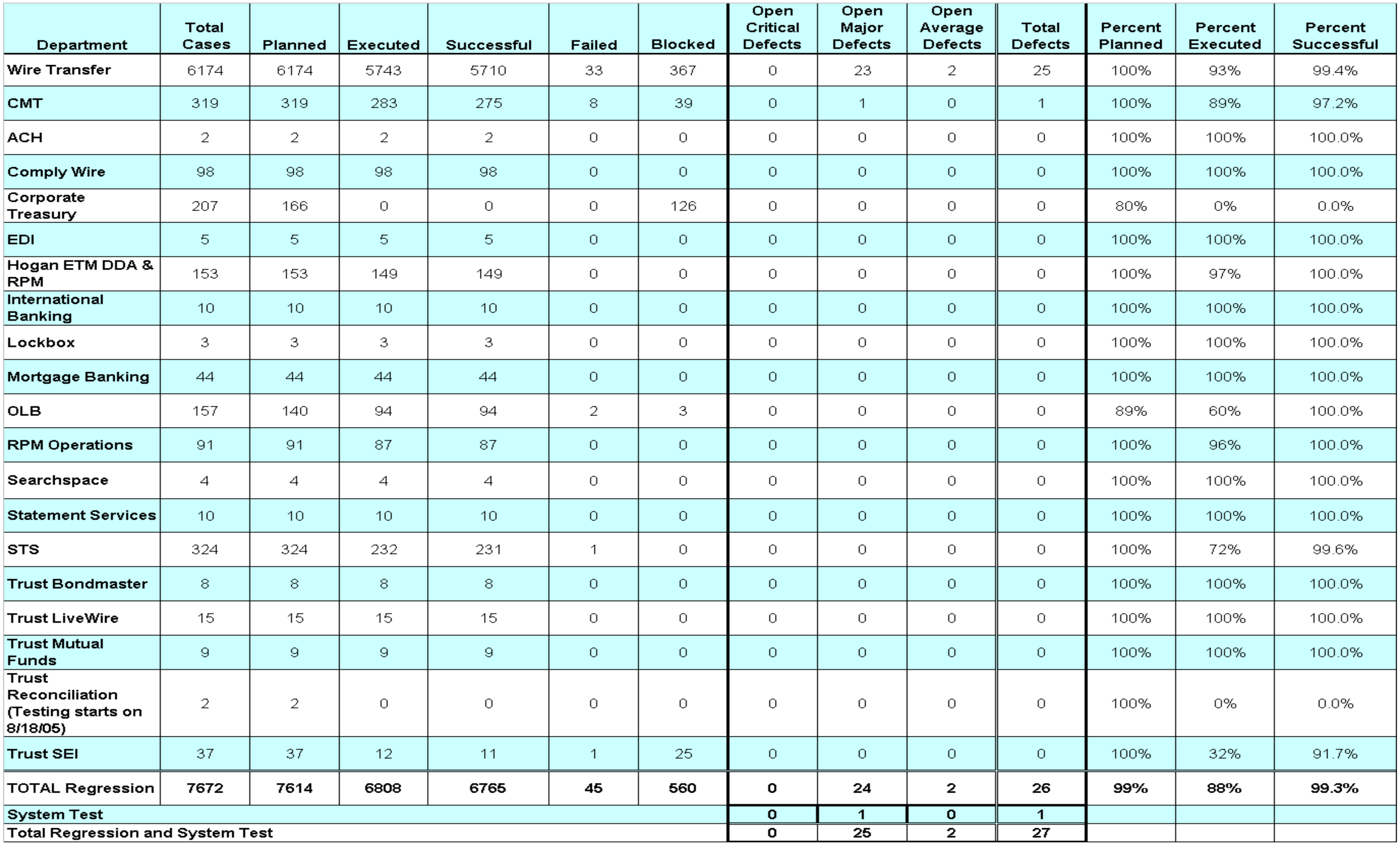
Key testing challenge during test execution is lack of time!

Example Test Cases Executed



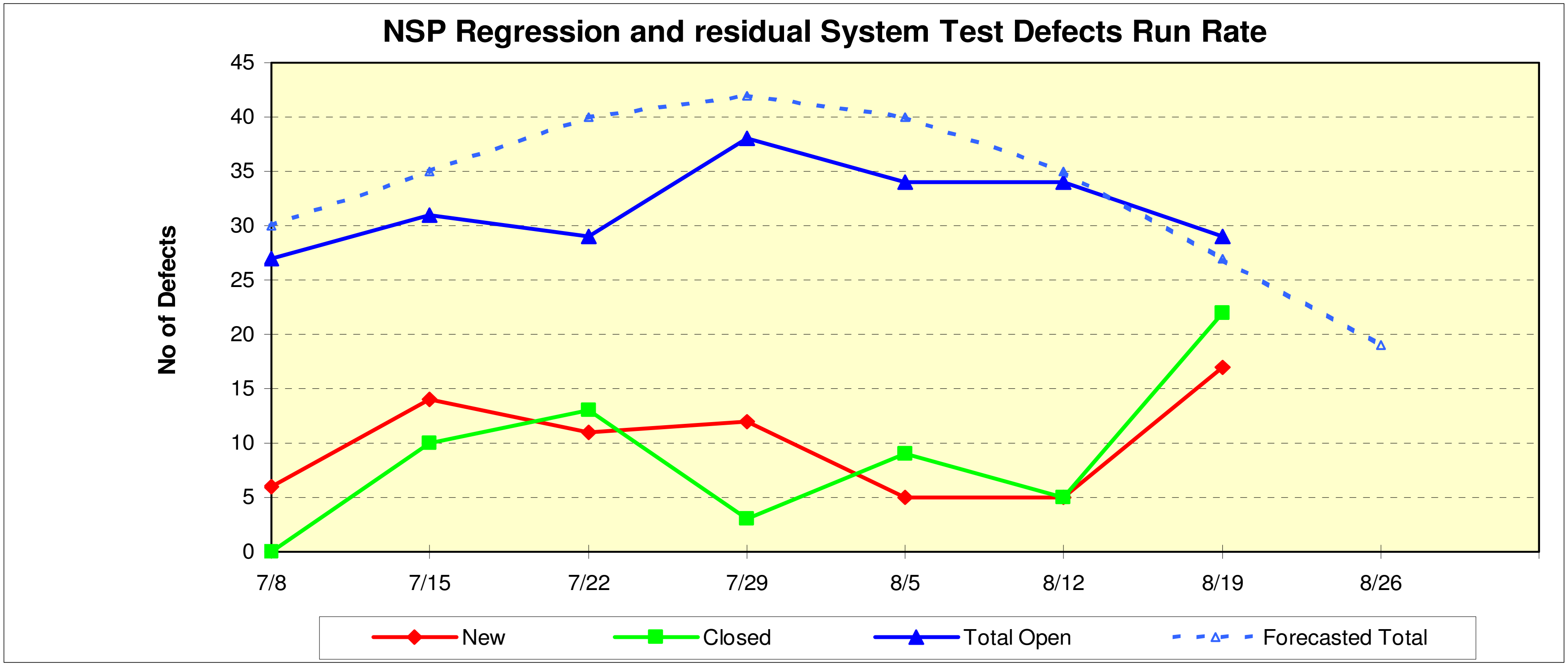
Date:	07/08/05	07/15/05	07/22/05	07/29/05	08/05/05	08/12/05	08/19/05	08/26/05
Total:	7672	7672	7672	7672	7672	7672	7672	7672
Planned:	575	1644	2668	3805	4973	6234	7584	7672
Planned %	7%	21%	34%	49%	64%	81%	98%	100%
Executed %	10%	29%	49%	62%	69%	76%	88%	
Success Rate %	99.2%	99.0%	99.3%	99.3%	99.3%	99.4%	99.3%	

Example Testing by Application Area



Department	Total Cases	Planned	Executed	Successful	Failed	Blocked	Open Critical Defects	Open Major Defects	Open Average Defects	Total Defects	Percent Planned	Percent Executed	Percent Successful
Wire Transfer	6174	6174	5743	5710	33	367	0	23	2	25	100%	93%	99.4%
CMT	319	319	283	275	8	39	0	1	0	1	100%	89%	97.2%
ACH	2	2	2	2	0	0	0	0	0	0	100%	100%	100.0%
Comply Wire	98	98	98	98	0	0	0	0	0	0	100%	100%	100.0%
Corporate Treasury	207	166	0	0	0	126	0	0	0	0	80%	0%	0.0%
EDI	5	5	5	5	0	0	0	0	0	0	100%	100%	100.0%
Hogan ETM DDA & RPM	153	153	149	149	0	0	0	0	0	0	100%	97%	100.0%
International Banking	10	10	10	10	0	0	0	0	0	0	100%	100%	100.0%
Lockbox	3	3	3	3	0	0	0	0	0	0	100%	100%	100.0%
Mortgage Banking	44	44	44	44	0	0	0	0	0	0	100%	100%	100.0%
OLB	157	140	94	94	2	3	0	0	0	0	89%	60%	100.0%
RPM Operations	91	91	87	87	0	0	0	0	0	0	100%	96%	100.0%
Searchspace	4	4	4	4	0	0	0	0	0	0	100%	100%	100.0%
Statement Services	10	10	10	10	0	0	0	0	0	0	100%	100%	100.0%
STS	324	324	232	231	1	0	0	0	0	0	100%	72%	99.6%
Trust Bondmaster	8	8	8	8	0	0	0	0	0	0	100%	100%	100.0%
Trust LiveWire	15	15	15	15	0	0	0	0	0	0	100%	100%	100.0%
Trust Mutual Funds	9	9	9	9	0	0	0	0	0	0	100%	100%	100.0%
Trust Reconciliation (Testing starts on 8/18/05)	2	2	0	0	0	0	0	0	0	0	100%	0%	0.0%
Trust SEI	37	37	12	11	1	25	0	0	0	0	100%	32%	91.7%
TOTAL Regression	**7672**	**7614**	**6808**	**6765**	**45**	**560**	**0**	**24**	**2**	**26**	**99%**	**88%**	**99.3%**
System Test							**0**	**1**	**0**	**1**			
Total Regression and System Test							**0**	**25**	**2**	**27**			

Example Weekly Defect Run Rate



Defect Rate								
Week Ending	**7/8**	**7/15**	**7/22**	**7/29**	**8/5**	**8/12**	**8/19**	**8/26**
New	6	14	11	12	5	5	17	
Closed	0	10	13	3	9	5	22	
Total Open	27	31	29	38	34	34	29	
Forecasted Total	30	35	40	42	40	35	27	19

Example Daily Defects - Regression & System Test

Defects by Development Team	Daily Results - Regression and System							
	8/11	8/12	8/15	8/16	8/17	8/18	8/19	8/22
Total Open	33	34	26	25	25	22	29	27
COPE/STAR	26	26	16	18	16	16	19	18
CIR	2	2	2	2	2	2	2	2
STS	1	1	1	1	1	0	0	0
USB Internal	2	4	3	3	1	0	2	5
Submitted	2	1	4	1	5	4	6	2
Total In Re-test	11	18	8	6	6	4	7	6

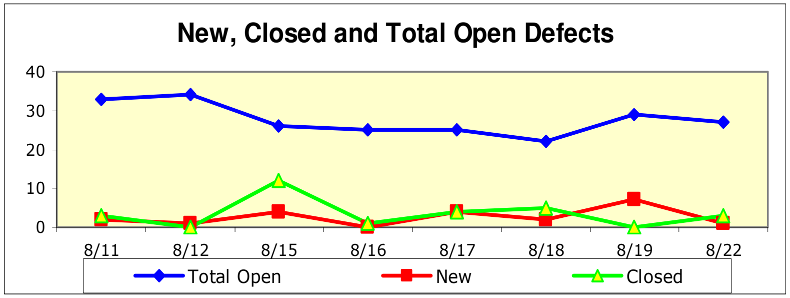
Defects by Environment	Daily Results - Regression and System							
	8/11	8/12	8/15	8/16	8/17	8/18	8/19	8/22
Total Open	33	34	26	25	25	22	29	27
Regression	27	28	24	23	23	21	28	26
System Test	6	6	2	2	2	1	1	1

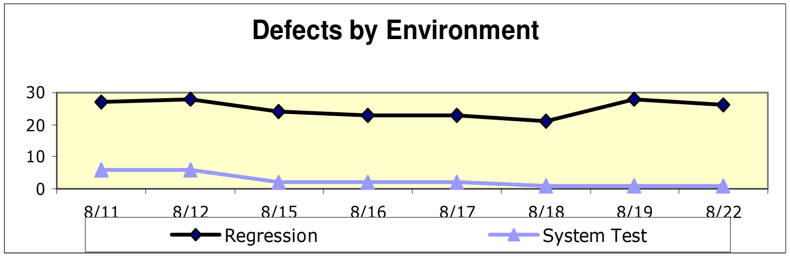
Defects by Severity	Daily Results - Regression and System							
	8/11	8/12	8/15	8/16	8/17	8/18	8/19	8/22
Critical	3	3	3	4	1	1	1	0
Major	26	27	21	19	22	20	27	25
Average	4	4	2	2	2	1	1	2
Total OPEN	**33**	**34**	**26**	**25**	**25**	**22**	**29**	**27**

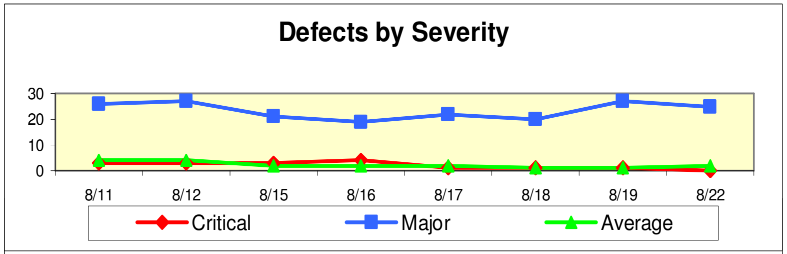
Defects - A.04 Validation	Daily Results - Regression and System								
	8/11	8/12	8/15	8/16	8/17	8/18	8/19	8/22	Total
New	2	1	4	0	4	2	7	1	21
Closed	3	0	12	1	4	5	0	3	28

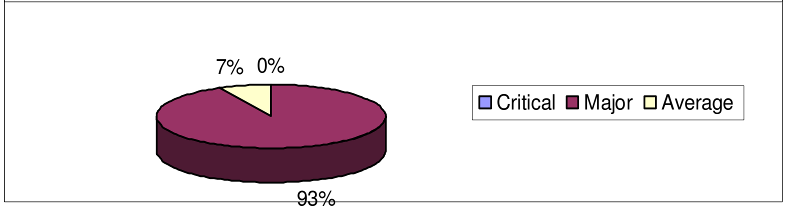
* Two defects moved to A.04a on 8/1

(8-day moving window)

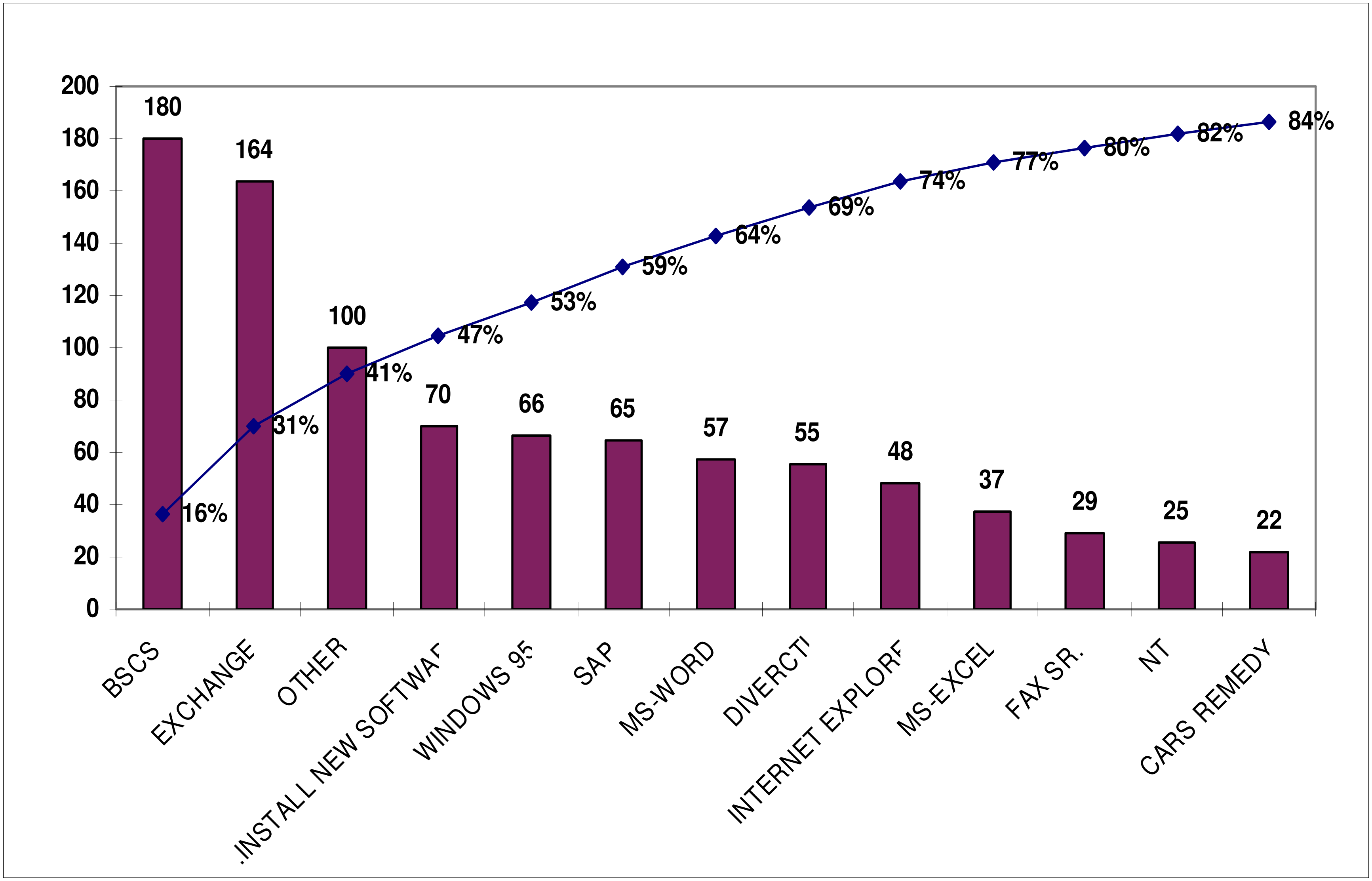




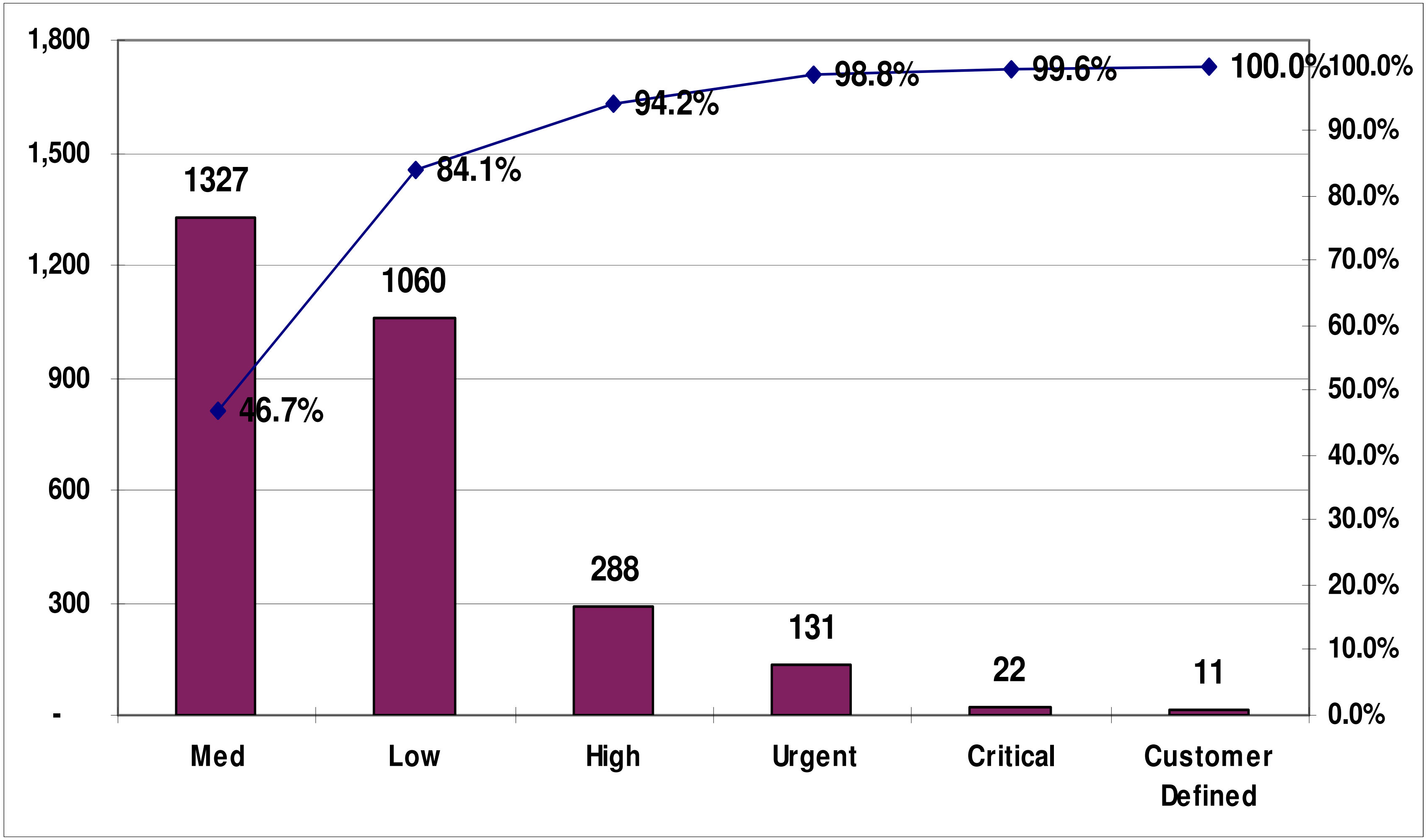




Defects by Category



Defects by Priority

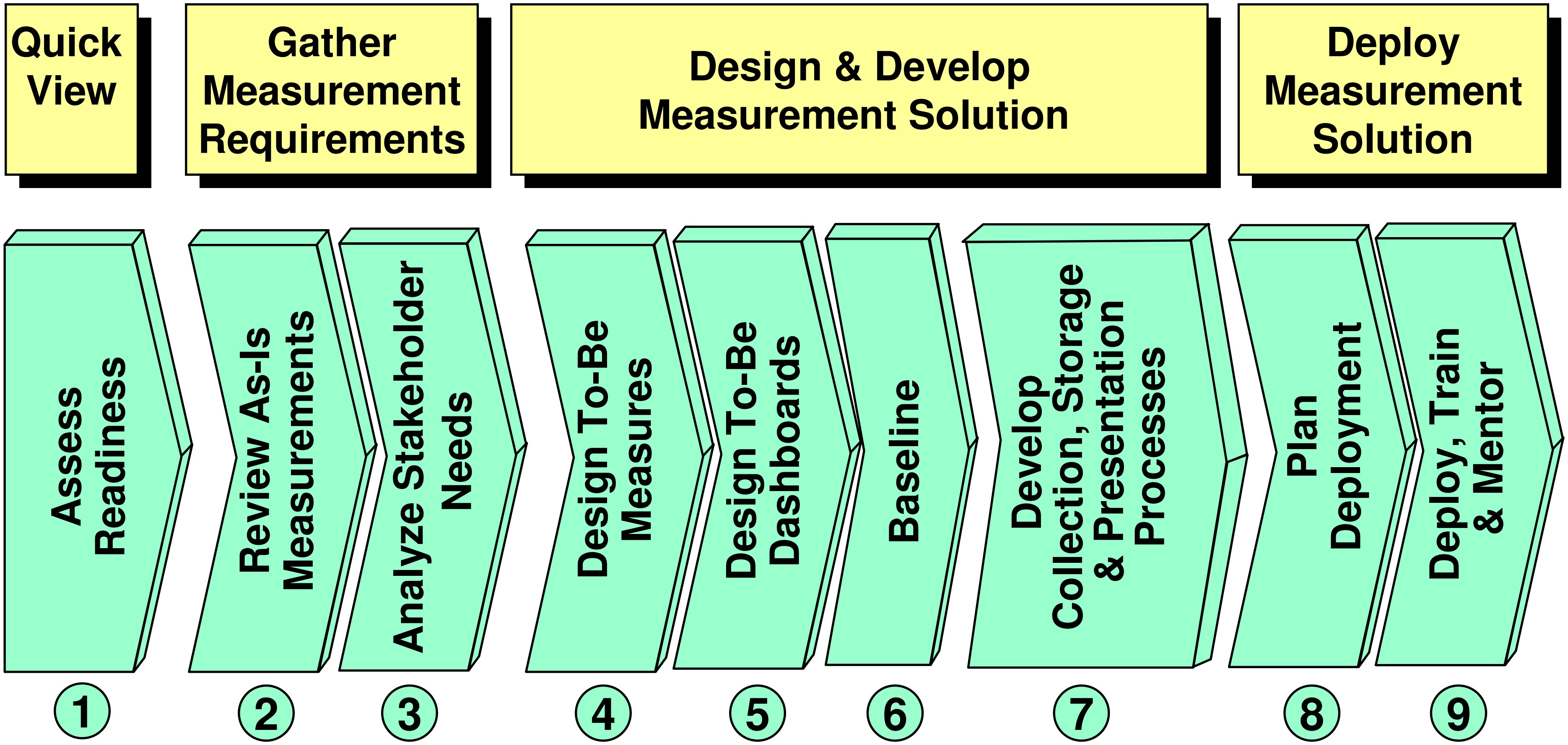


Agenda

- Testing Process Improvement
- Linking Process Improvement and Measurement
- Testing Process and Metrics
- Testing Measurement Program Design
- Summary

Designing a Measurement System

- Run Testing Measurement like a Project
- Follows a similar approach to application development



Quick View

Gather Measurement Requirements

Design & Develop Measurement Solution

Deploy Measurement Solution

1 Assess Readiness

2 Review As-Is Measurements

3 Analyze Stakeholder Needs

4 Design To-Be Measures

5 Design To-Be Dashboards

6 Baseline

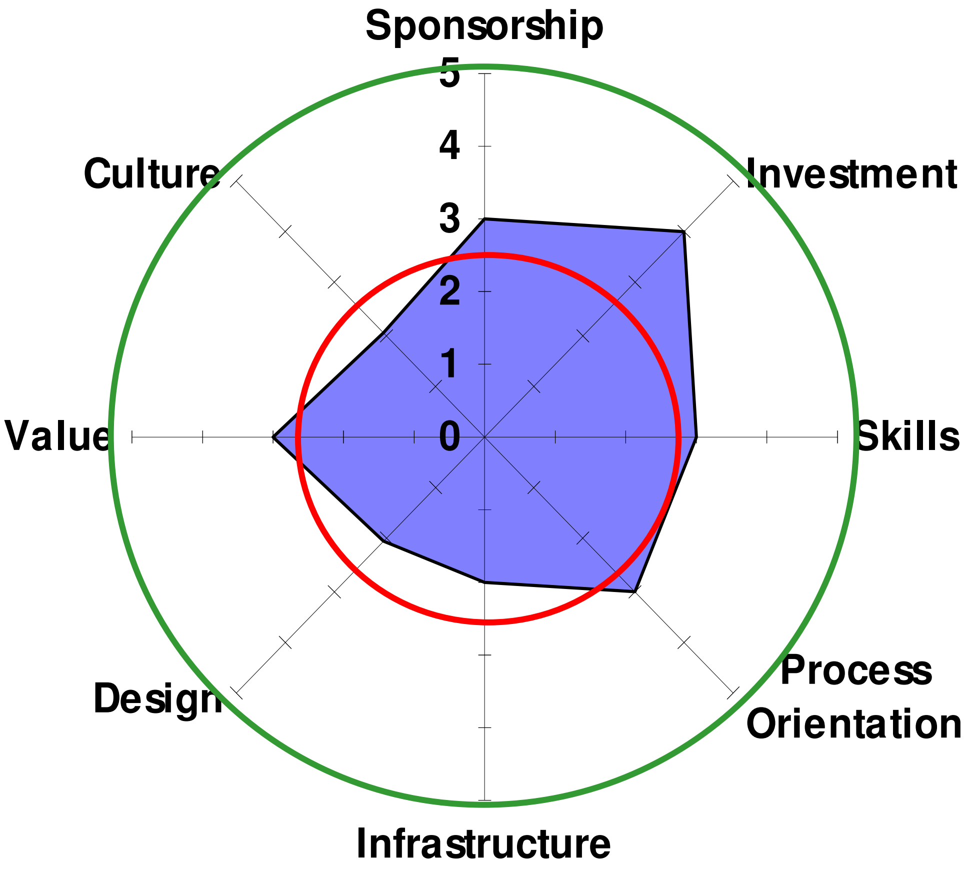
7 Develop Collection, Storage & Presentation Processes

8 Plan Deployment

9 Deploy, Train & Mentor

1 Assess whether Organization is Ready for Measurement and Process Improvement

A Readiness Survey can be used to gather of information about the organization’s readiness.



2 Review As-Is Measures

Identify measures that are being collected today

- ♦ How are they being collected
- ♦ How are they defined
- ♦ How are they reported
- ♦ How are they organized into Dashboards, Reports, etc
- ♦ Who is receiving measures
- ♦ What do they use them for
- ♦ Ask about past efforts at introducing measurement
- ♦ Look at measures from other groups within organization

③ Analyze Stakeholder Measurement Needs

- Sr. management
- Middle Management
- Peer Managers
- Direct reporting staff
- Key functional groups that are suppliers or customers:
  - Sales
  - Manufacturing
  - or Finance

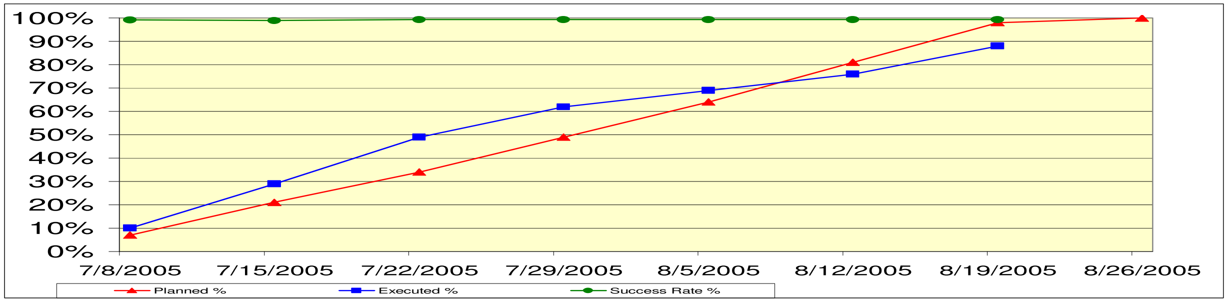
Don’t assume that stakeholders know what they want. Use their input as a starting point.

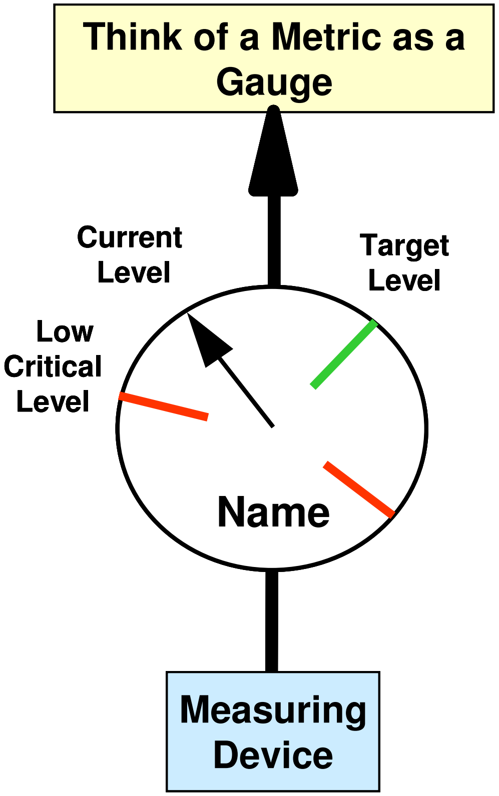
④ Design the To-Be Measures

- **Start with As-Is measures & stakeholder needs**
- **Use Reference measures:**
  - **Benchmark Studies**
  - **Other business groups, agencies, user group, societies and consulting companies**
  - **Books on measurement such as: Practical Software Measurement**
- **Use proven techniques to develop measures:**
  - Balanced Score Card
  - Goal->Question->Metric

**We covered a number of testing measures earlier.
Although some measures are better than others, there is no single "right" answer – Just Do It! And continuously improve!**

4 Document Measures in a Measurement Catalog

Term	Definition
Name	Measurement Name
Graph	Graph of Measure
	
Scale	Unit of Measure
Frequency	Frequency of data collection and reporting
Audience	Who should see & use measure
Measuring Device	Mechanism used to collect data
Current Level	Current level of measure
Critical Level	Anything worse (High, Low or Both) indicates failure
Target Goal	Target improvement level & time frame
Meaning	Simple definition of what a High, Low or Tend may mean.
Comments	Comments, references, examples, etc.



Think of a Metric as a Gauge

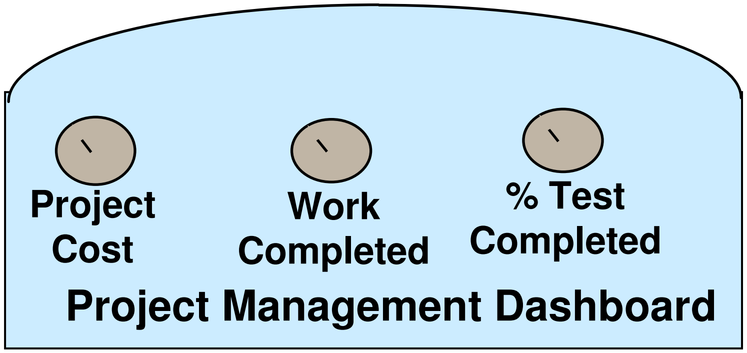
Measuring Device

4 Set SMART Measurement Goals

- **S**pecific – valid measures for the thing that is being measured
- **M**easurable – measurement data must be collected:
  - Accurately
  - Precisely
  - In a way that is repeatable
  - In a way that can be reproduced
- **A**ctionable –metrics must be easy-to-understand and easy to use when making decisions
- **R**elevant – measure things that are important
- **T**imely – provide data when it is needed

5 Design Dashboards & Score Cards

- Organize metrics into graphical views of related measures to enable stakeholders to see interdependencies
- Different views may be required for different stakeholders
  - Directors of Development
  - Test Managers
  - Project Managers
- The same measure may occur on multiple dashboards

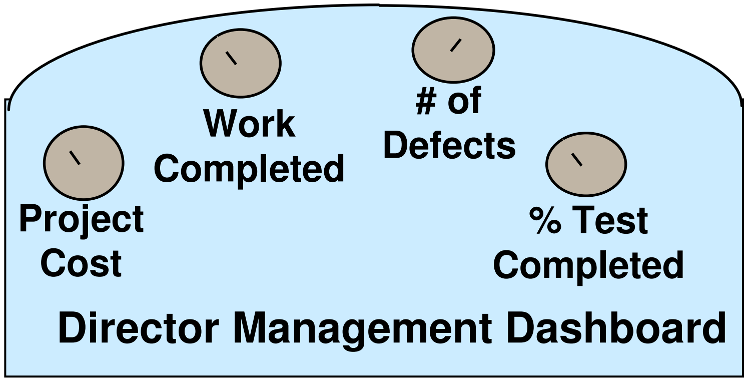


Project Cost

Work Completed

% Test Completed

Project Management Dashboard



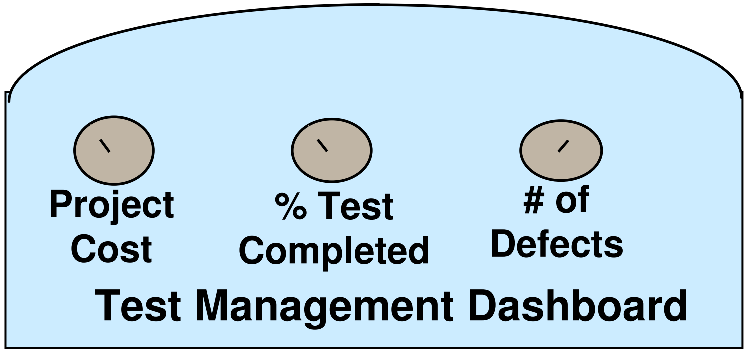
Work Completed

# of Defects

Project Cost

% Test Completed

Director Management Dashboard



Project Cost

% Test Completed

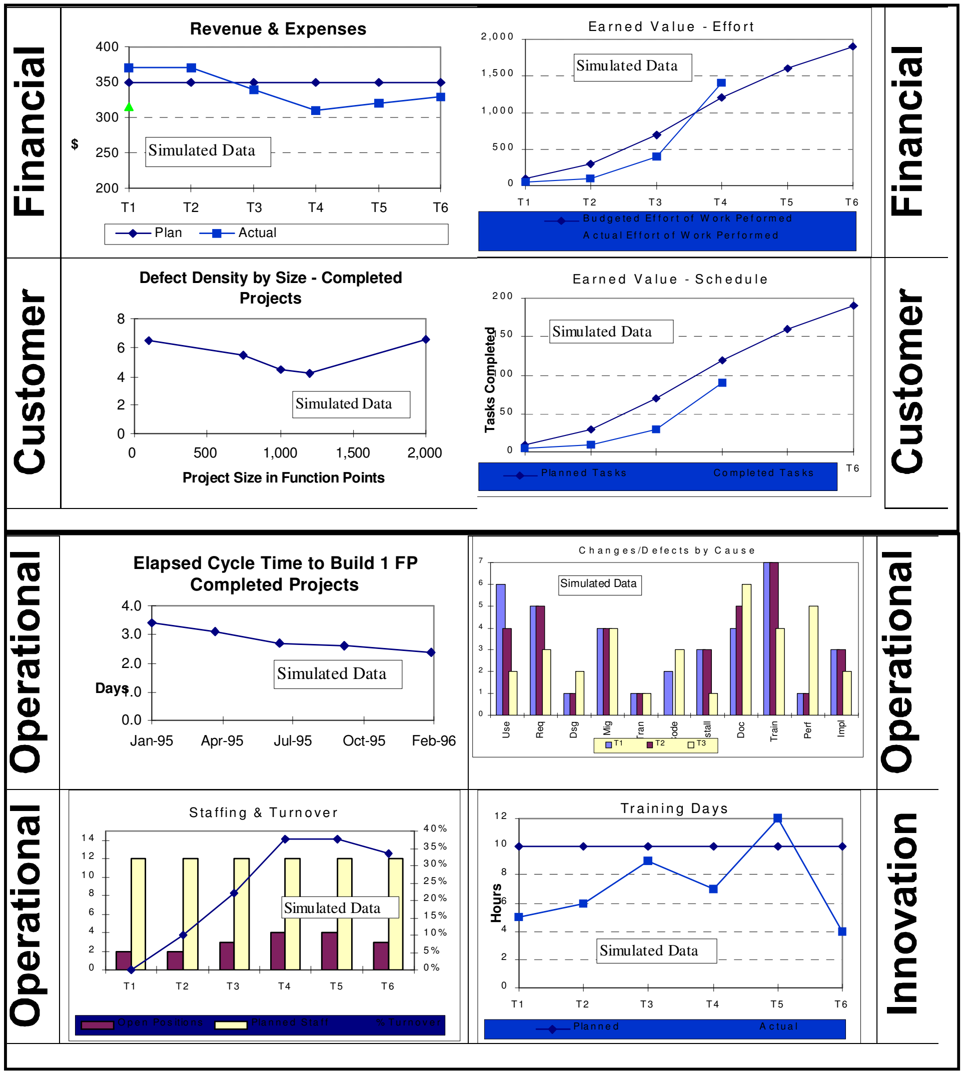
# of Defects

Test Management Dashboard

Score Cards tend to be linked to organizational goals. *Dashboards* tend to be focused on operational controls. In general the terms can be used interchangeably

5 Example Balanced Score Card

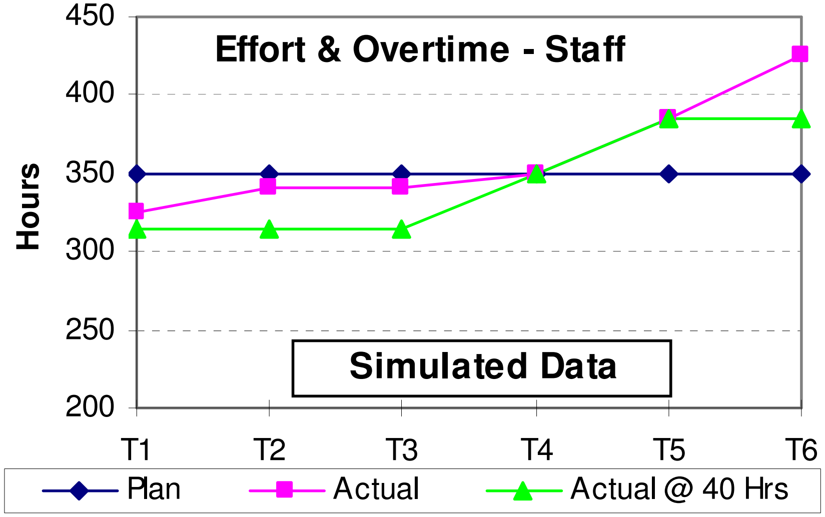
- **Financial:**
  - Revenue vs. Expense
  - Earned Value - Effort
- **Customer:**
  - Defect Density
  - Earned Value - Schedule
  - Customer Satisfaction
- **Operational:**
  - Cycle Time - Days per FP
  - Defects by Cause
  - Turnover
- **Improvement:**
  - Training Days
  - SEI Maturity Level

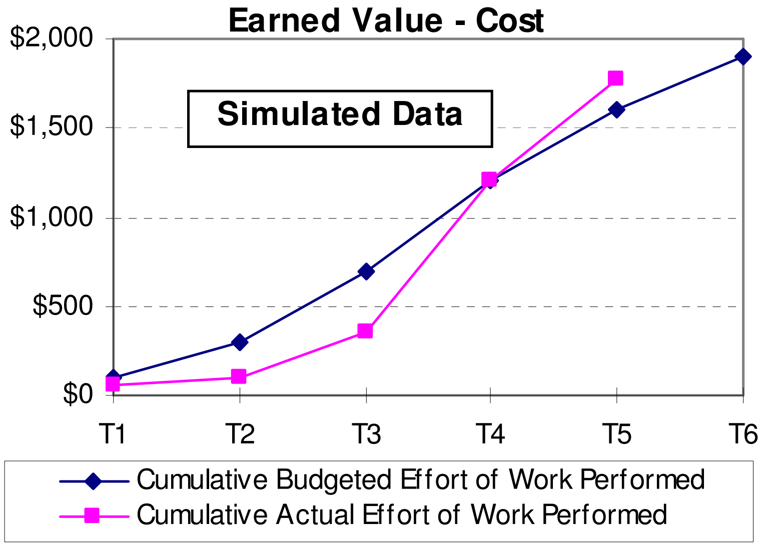


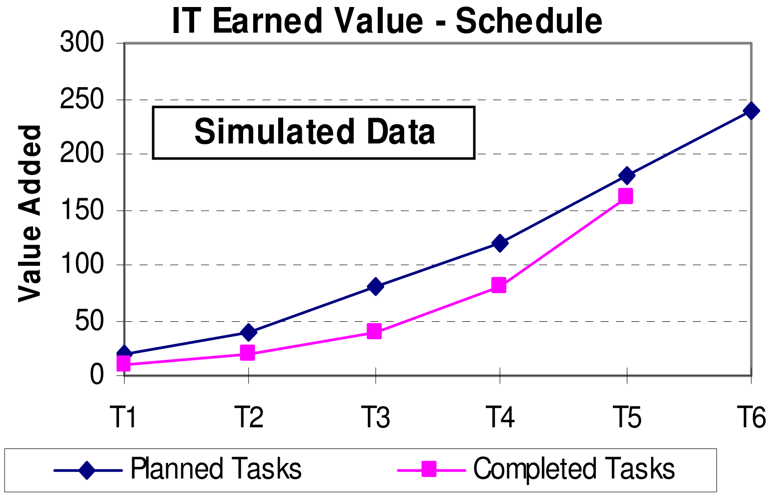
Measurement Dashboard Sample

5 Measures are often Interdependent

- Test Project starts out understaffed
- Test Manager catches up via overtime and additional resources
- Project is getting back to plan
- But is starting to run over budget because of the overtime and extra staff





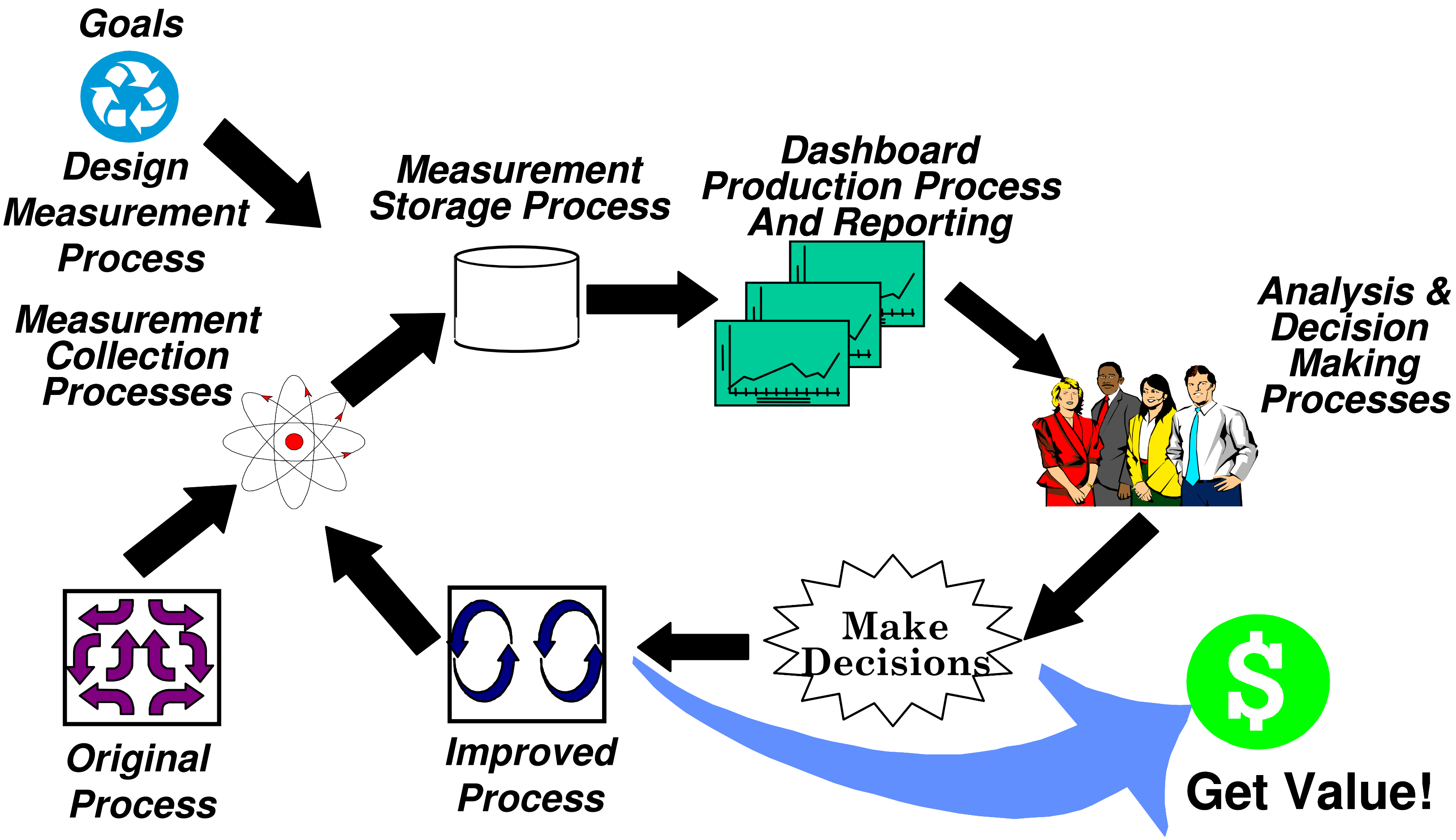


6 Baseline – Required for Process Improvement

- Differences between Baselines show progress
- Perform a Measurement Baselines before a process improvement
- Perform additional Measurement Baselines during and after process improvement
- Use historical data coupled with new measurement collection processes to gather baseline data

A Measurement Baseline involves measuring the values for measures at a specific point in time.

7 Develop a Solid Measurement Infrastructure


Goals
Design Measurement Process
Measurement Storage Process
Dashboard Production Process And Reporting
Analysis & Decision Making Processes
Measurement Collection Processes
Make Decisions
Original Process
Improved Process
Get Value!

The Measurement Infrastructure includes all components necessary to consistently collect, store, analyze and present metrics to stakeholders on a regular basis.

⑧ Plan Measurement Deployment

- Which measures to develop?
- How they will be collected, stored and analyzed?
- What Dashboards & Score Cards are needed?
- Who will develop the operational reports, Dashboards & Score Cards?
- How will the Dashboards and Score Cards be distributed?
- Who needs to be Educated?
  - Which stakeholders?
  - What to educate them on?
  - Do not assume that stakeholders will know what to do with the measures once they get them

In order to use measures to manage their business, Stakeholders must be mentored over time in the use of the measures

9 Deploy, Educate, Mentor & Refine

- Deploy - Collect, store and produce dashboards on a periodic basis (set realistic targets)
- Educate stakeholders on the measures
- Provide a help desk to answer metrics questions
- Mentor stakeholders in the use of measures to solve problems
- Continuously improve

Use Them or Lose Them!

Use measurement to answer questions and to solve problems or the measurement program will be cancelled!

Agenda

- ♦ Testing Process Improvement
- ♦ Linking Process Improvement and Measurement
- ♦ Testing Process and Metrics
- ♦ Testing Measurement Program Design
- ♦ **Summary**

Summary – Metrics for Testing Excellence!

- Measurement goals must support organization processes & goals
- The organization must be ready for measurement
- Select the right things to measure
  - Use the Balanced Score Cards to link measures to your goals
  - Use Goal->Question->Metric to identify key metrics
- Organize measures into dashboards
- Develop a solid Measurement Infrastructure
- Measure processes not individuals
- Start small and stick with it
- Mentor stakeholders – they must use the measures for the program to be successful

Build an excellent Testing Organization by developing an excellent measurement program!

The Benefits are Worth It

Benefit	Level 1 Baseline	Level 2	Level 3	Level 4	Level 5
Rework	40%	20%	10%	6%	3%
Estimating accuracy	20% to >100%	±10% to ±20%	±5%	±3%	±1%
Delivered defects	X	½ X	1/4 X	1/10 X	1/100 X
Productivity	X	1.5X	2X	3-4X	>4X

Source: Data extrapolated from published case studies of benefits from CMM/CMMI improvement programs – Presented by Bill Curtis Presentation at the ITMPI Conference in Chicago, Nov. 13, 2008

Questions and Discussion

Larry Dribin, Ph.D.

larry@pearlstreetgroup.com

847-807-7390