



Software Testing and the Capability Maturity Model

Jim D. Hart

e-mail: jimdhart@idc-inc.com

Principal Consultant

Innovation Dynamics Consulting, Inc.

(704) 846-5853 <or> <http://www.idc-inc.com>



Agenda

Overview of the Capability Maturity Model for Software (CMM)

- A definition for **software process**
- The focus on process
- The five levels of the CMM

Testing at each level of maturity

- **Visibility of process**
- **Process characteristics**
- **Effects of processes**



Agenda

Overview of the Capability Maturity Model for Software (CMM)

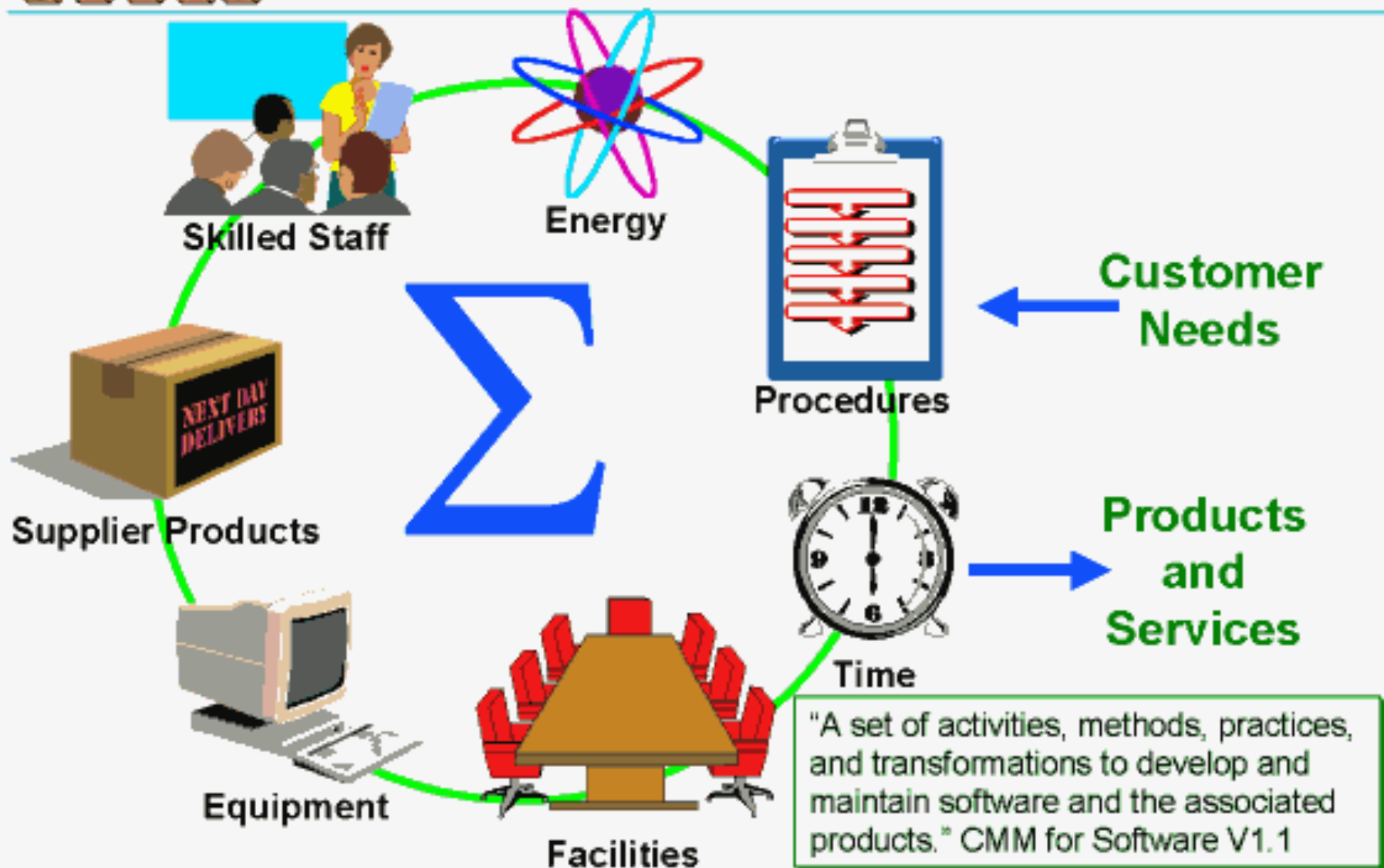
- A definition for **software process**
- The focus on process
- The five levels of the CMM

Testing at each level of maturity

- Visibility of process
- Process characteristics
- Effects of processes



Definition of Software Process





Why Focus On Process?

The quality of a product or service is governed primarily by the quality of the process used to develop and deliver the product or service.

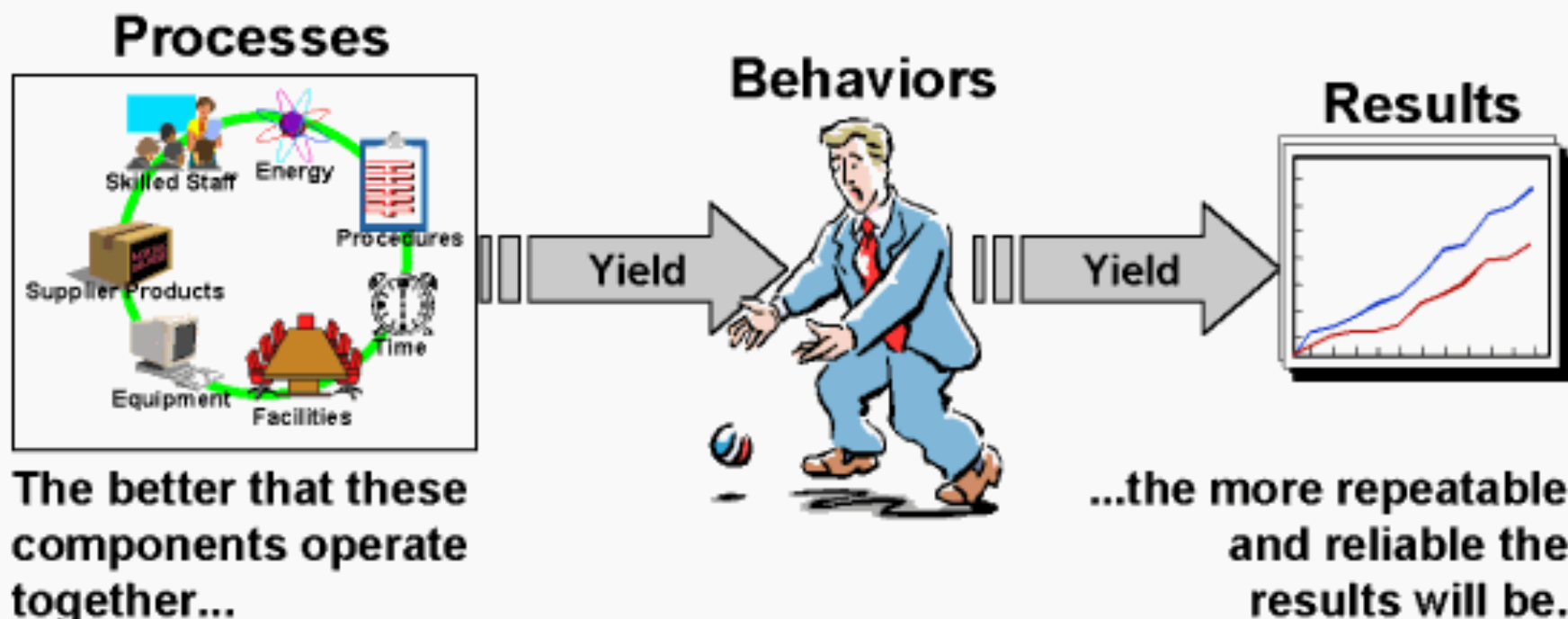
This premise implies focus on managing the process as well as managing results.

Manufacturing and service industries world-wide have demonstrated the value of this premise for over 50 years.

Many software organizations are now applying this premise to software development and maintenance.



Effects of Process Maturity





Evolution of Process Maturity

Level	Process Description	Process Attributes in Place
5 Optimizing	Processes are continuously examined and improved.	Automated: Manual steps replaced Improved: Opportunities tested Error-proofed: Causes for defects removed
4 Managed	Product and process quality is quantitatively measured and controlled.	Streamlined: Remove no-value activities Controlled: Kept within quality limits
3 Defined	Technical and management practices are integrated into a well-defined process.	Measured: Quantitatively understood Tailored: To meet specific customer needs Standardized: Consistently followed
2 Repeatable	Project management practices are mastered and repeatable.	Practiced: Followed by staff Trained: Staff have the skills Documented: Practices are written down
1 Initial	Processes are unpredictable and poorly controlled.	Ad hoc: Everyone does it differently



Agenda

Overview of the Capability Maturity Model for Software (CMM)

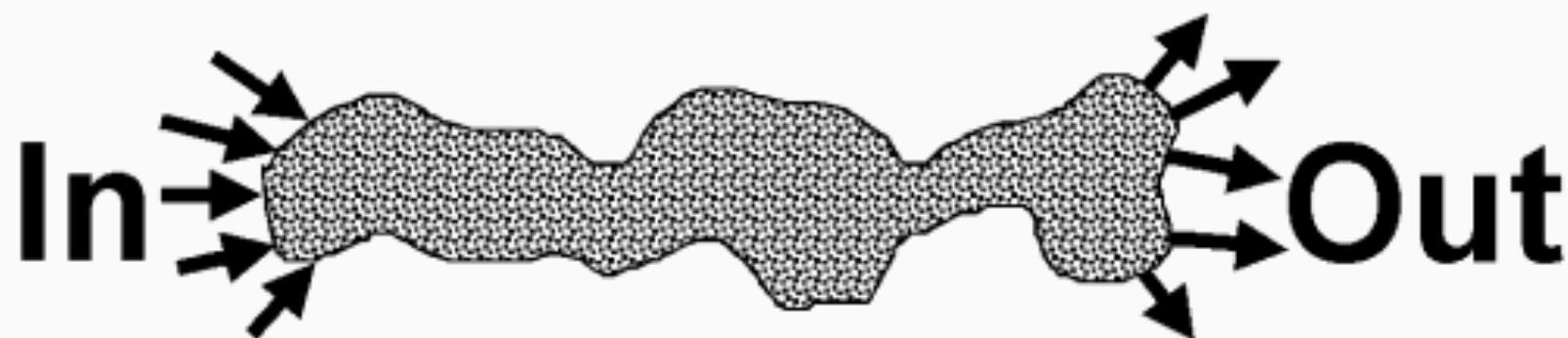
- A definition for software process
- The focus on process
- The five levels of the CMM

Testing at each level of maturity

- **Visibility of process**
- **Process characteristics**
- **Effects of processes**



Visibility into the Process at Level 1



Requirements flow in uncontrolled and unpredictably.

A software product is (usually) produced by a process that is not well defined or commonly understood.

“Products are not released, they escape.”

Source for graphic: CMU/SEI-93-TR-25



Testing at the Initial Level

Process characteristics:

- **Testing activities are usually part of the development process. Effort and duration estimates for testing are often either unrealistic or not incorporated into the project plans.**
- **Test cases are seldom planned in advance of testing (especially functional testing activities).**
- **Defects are seldom recorded.**
- **System-level testing usually replicates user acceptance testing in order to remove defects the customer would normally find.**
- **Phases and types of testing are not explicitly managed (controlled). Testing levels often overlap.**



Testing at the Initial Level

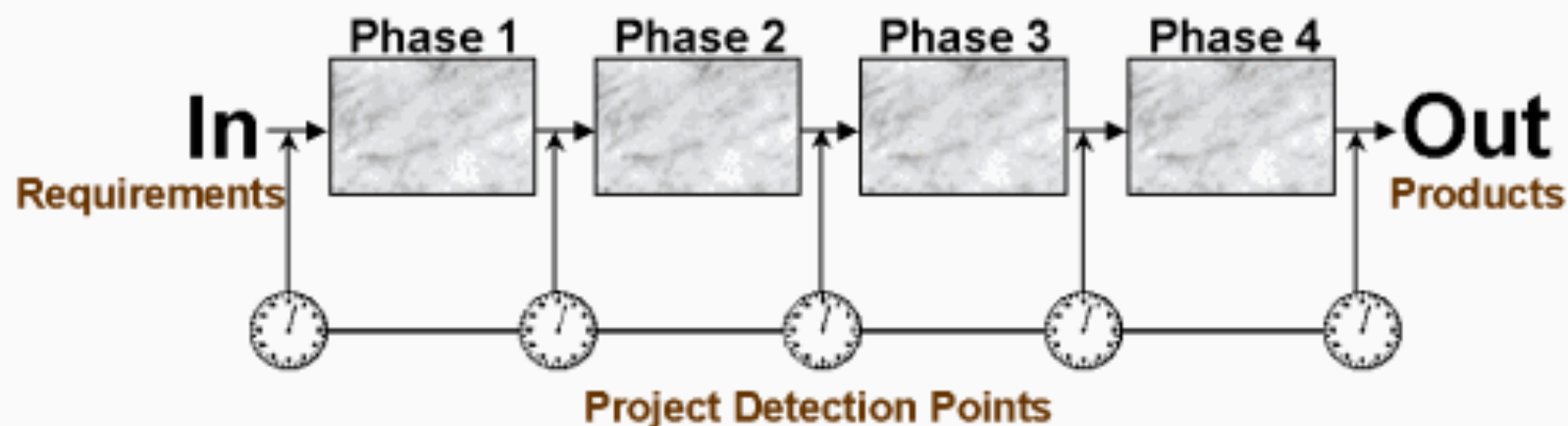
Effects:

- **There is a general lack of a disciplined testing process.**
- **The Test Squeeze:**
 - **Effort and duration estimates for conducting testing activities are often over-optimistic.**
 - **Testing seldom begins when planned, but is often “completed” on schedule.**
 - **Testers test until time expires.**
- **Little if any defect data is recorded as there is seldom enough time. Instead, defects are generally fixed on the spot before they are forgotten.**



Visibility into the Process at Level 2

Software Project Life Cycle



Requirements, resources, and artifacts are controlled.

Software production is planned, and tracked at pre-defined points in the project.

Procedures exist for fundamental project management activities. There are no procedures necessary for testing at this level of maturity.



Testing at the Repeatable Level

Process characteristics:

- **Test “phases” are defined through the projects’ selected software life cycles.**
- **Testing activities are planned.**
 - **Testing activities are “sized.” Size estimates are used to derive estimated effort (e.g., staff-hours), duration, staffing, and schedule.**
 - **Testing activities are estimated based on historical data or experiential knowledge.**
 - **Those performing testing participate in establishing project commitments.**
- **Change control is in place for fixing defects during formal testing.**



Testing at the Repeatable Level

Process characteristics (continued):

- **Testing activities are tracked.**
 - **Engineering milestone reviews are held with the customer to determine readiness for testing.**
 - **Testing status is measured, tracked to plans, and reported to management.**
 - **Project commitments (e.g., costs, staffing, delivery milestones, functionality) are renegotiated with the customer if actual test size, effort, or durations vary from the estimates.**
 - **Schedules and testing milestones are renegotiated if they are not likely to be met.**
- **Formal testing is usually done against software product baselines.**



Testing at the Repeatable Level

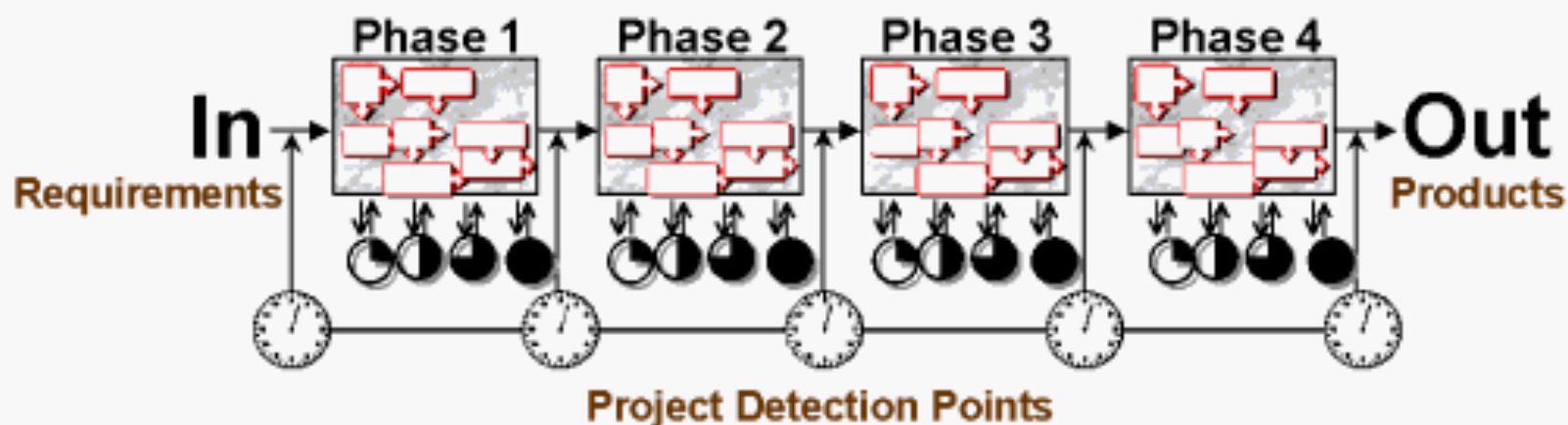
Effects:

- **In general, there are adequate time, resources, and staff assigned to manage testing activities.**
- **Projects are delivered when scheduled testing activities have been completed.**
- **Automated test tools seldom succeed in having a measurable improvement in testing efficiency.**



Visibility into the Process at Level 3

Software Project Life Cycle



Roles and responsibilities in the process are understood. People have the necessary skills to perform their assigned tasks.

Software tasks are formally characterized, standard, visible, and managed throughout the software project.

Source for graphic: CMU/SEI-93-TR-25



Testing at the Defined Level

Process characteristics:

- Independent (skilled) test groups develop test plans, cases, and procedures.
- Projects implement explicit “levels of testing.”
 - **Unit testing** of software components
 - **Integration testing** of integrated software components against the software architecture
 - **System testing** of the system against the system and software requirements by the project
 - **Acceptance testing** of the system against system requirements by the customer/user
- Regression testing is planned and performed during each level of testing, based on project risks and historical defect data.



Testing at the Defined Level

Process characteristics (continued):

- **The organization establishes and uses standardized test processes.**
- **Projects define and enforce readiness and completion criteria for each level of testing.**
- **Test plans are derived from risk management plans and traceability matrices (mapping system requirements to software requirements to design to code).**
- **Staff keep test tools, scenarios, cases, procedures, scripts, harnesses, and results under configuration management.**



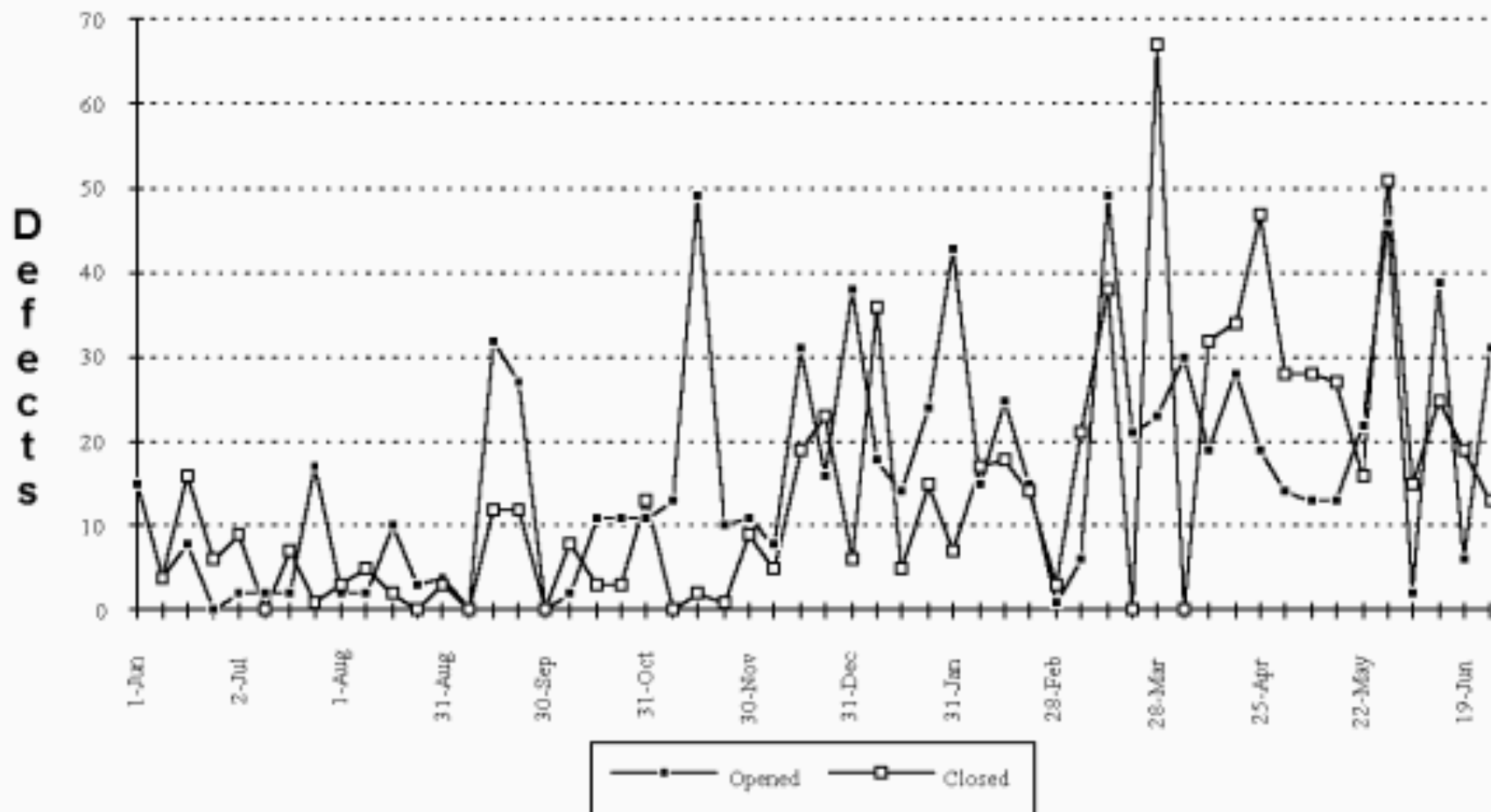
Testing at the Defined Level

Process characteristics (continued):

- **“Defect” is defined in measurable terms.**
 - **Product inconsistencies against documented functional requirements.**
 - **Products’ inability to satisfy undocumented customer requirements.**
- **Projects use a defect management process to:**
 - **log defects found during testing**
 - **track action items to closure**
 - **close action items based on validating the fix**
- **Projects track test results through the life of the project (e.g., defects open, opened, and closed per unit of time).**



Sample Defect Run Chart





Testing at the Defined Level

Effects:

- **Fewer defects are found during testing activities.**
 - Institutionalized peer reviews (inspections) find some defects early.
 - Tighter controls on changes to requirements means fewer defects introduced late in the project.
 - More formalized engineering processes focus on quality requirements and design documents.
 - Traceability of changes focus testing activities on impact areas.
- **There is a significant reduction in the time spent conducting system and user acceptance testing.**



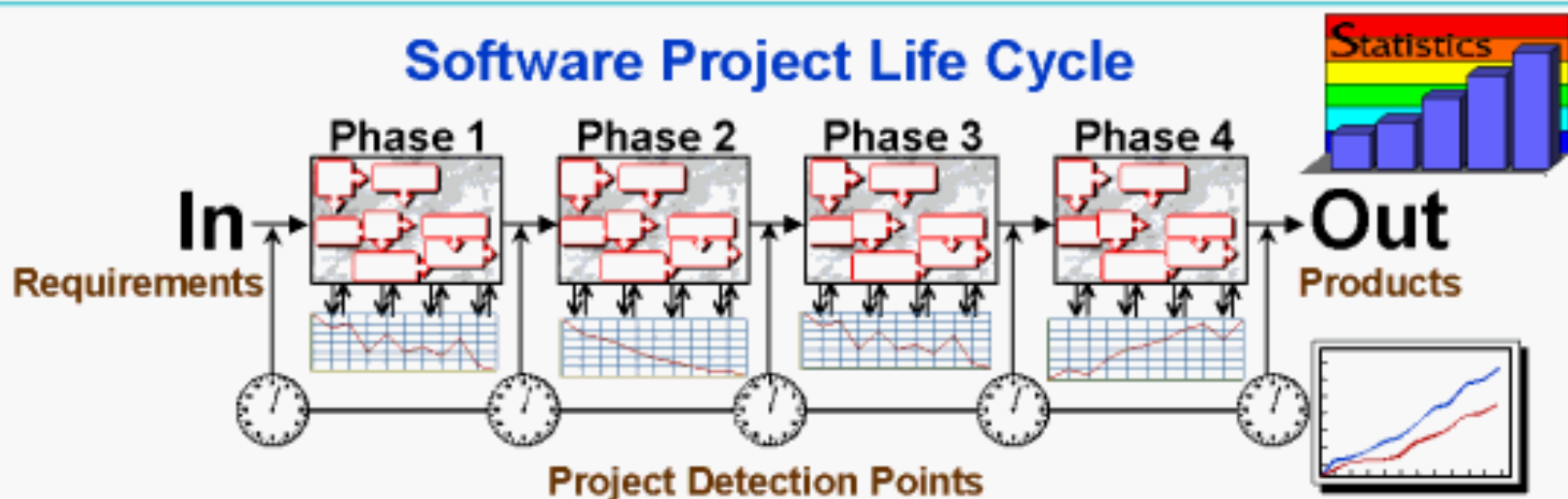
Testing at the Defined Level

Effects (continued):

- **Testing process efficiency and effectiveness are known (measured and analyzed), and used for future project planning.**
- **Standardized test processes increase repeatability of testing results. Testing processes are improved.**
- **Test results (e.g., defects open, opened, and closed per unit of time) are used to predict project milestone completions.**
- **Organizations can analyze and acquire automated test tools and equipment to support and speed up the testing processes.**



Visibility into the Process at Level 4



The production of the software product is quantitatively understood throughout the software process.

Management is able to predict software process performance within quantified bounds.

Source for graphic: CMU/SEI-93-TR-25



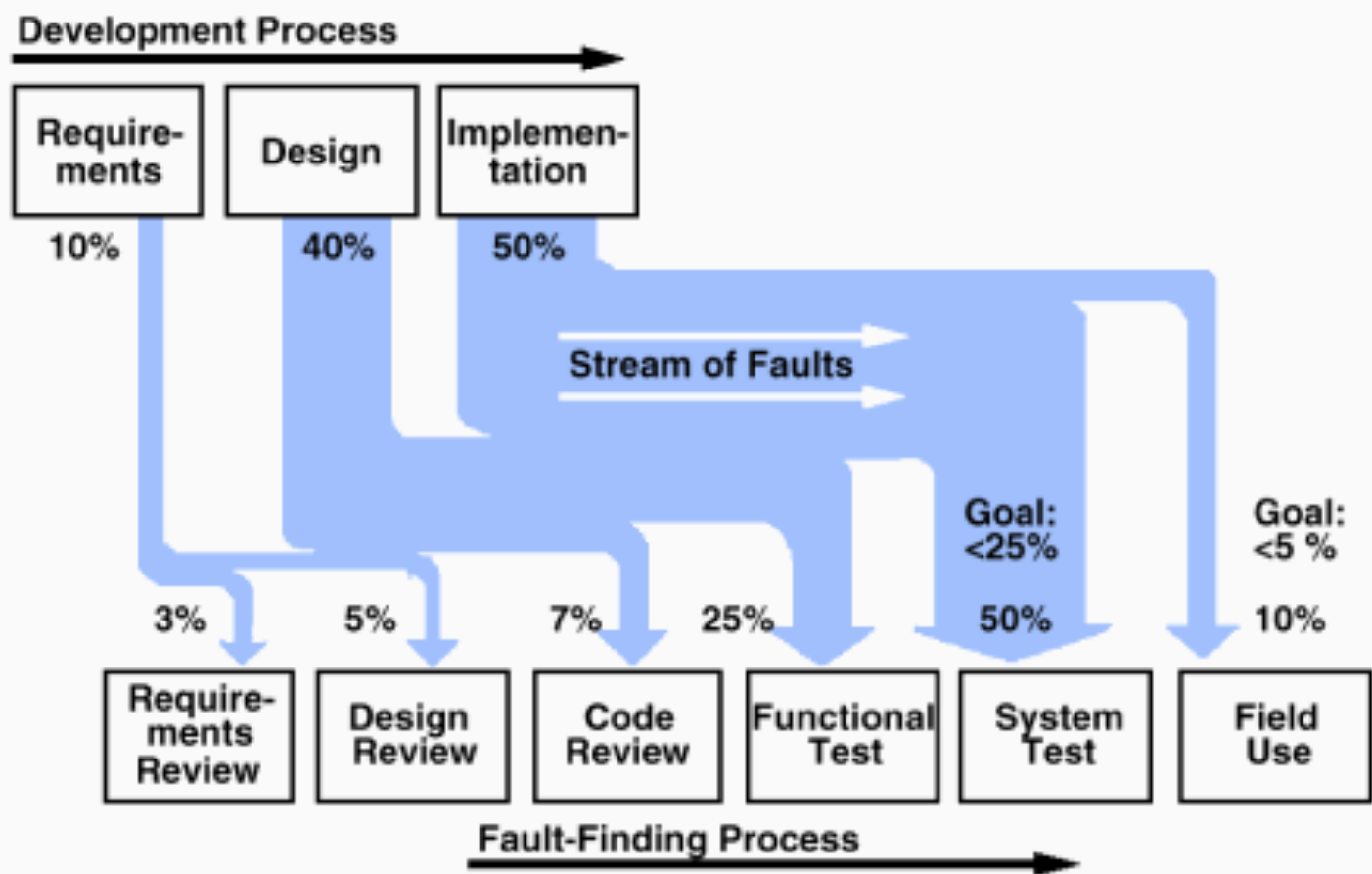
Testing at the Managed Level

Process characteristics:

- **Product quality is quantitatively measured during all testing activities.**
- **Engineers log and track defects found during functional testing activities (i.e., unit and integration levels) to closure.**
- **The quality of engineering processes are quantitatively measured and controlled based on defect data.**
 - **Historical defect densities (inherent errors per unit of product size) are used to estimate project defect densities.**
 - **Defect densities are used to estimate the effort and areas of testing activities.**



Fault Injection Model



Source: CMU/SEI-97-HB-003



Testing at the Managed Level

Process characteristics (continued):

- Product “quality” is defined in broader terms than just defects; for example,
 - **Maintainability** - the ease to which the software can be modified or fixed
 - **Reliability** - the probability that the software will continue operating when errors occur
 - **Reusability** - the ease to which the software can be used in other application development
 - **Portability** - the ease to which the software can be ported to another platform



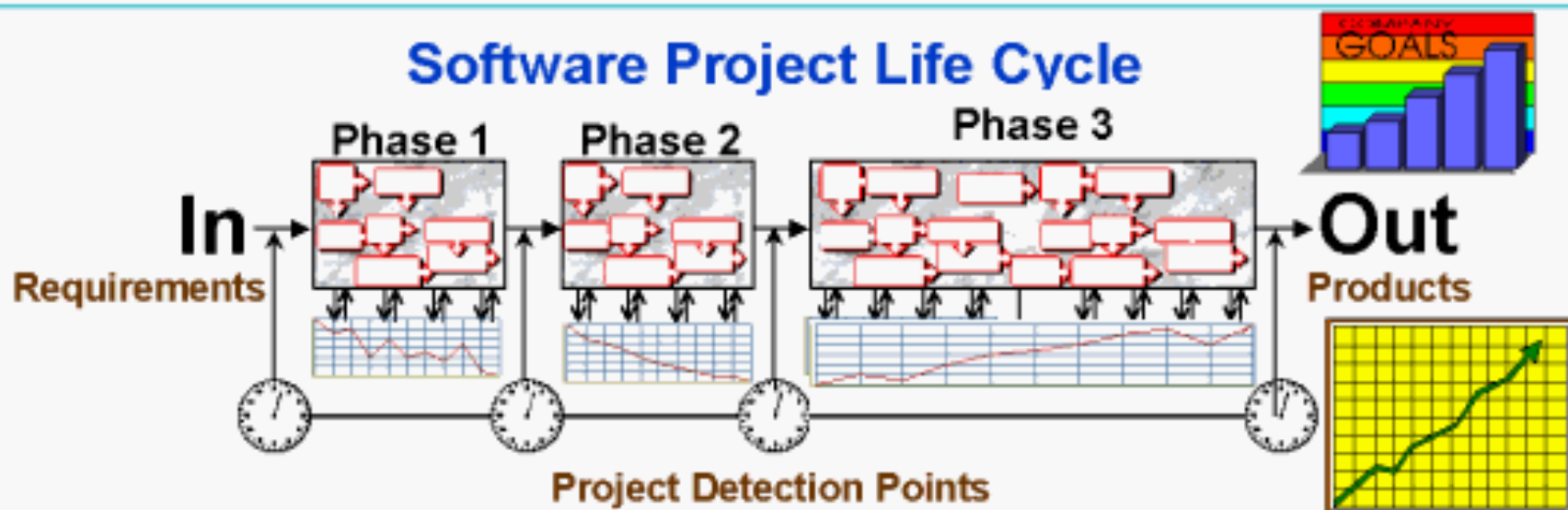
Testing at the Managed Level

Effects:

- **Fewer defects are found during later testing activities.**
 - **Most defects are found during peer reviews and early (functional) testing activities.**
- **Testing efficiencies (the effort expended per test conducted) and effectiveness (percentage of latent defects found) are understood and controlled.**
- **Test completion criteria can be based on quantitative data from tests conducted.**



Visibility into the Process at Level 5



The software process is continuously improved in a controlled manner.

Source for graphic: CMU/SEI-93-TR-25



Testing at the Optimizing Level

Process characteristics:

- **Engineering activities are modified to prevent defects from being built into the work products.**
 - **There is extensive focus on correcting engineering processes that inject defects.**
 - **Testing effectiveness is often used as a basis for improving testing processes.**
- **Automated test tools replace previously manual testing practices.**
- **The organization pilots new or revised testing processes and tools to determine their impacts on testing efficiency and/or effectiveness.**



Testing at the Optimizing Level

Effects:

- **Fewer defects are found during testing activities.**
 - Improved engineering activities prevent defects from being built into the work products.
 - Prototyping activities help eliminate potential requirements and design flaws.
- **Entire testing levels (e.g., user acceptance testing) may be dropped or combined with others.**
- **Testing efficiency and effectiveness are improved.**



Presentation Summary

- **The CMM is based on many years of practical experience in implementing mature software processes. It can help in improving your processes.**
- **High maturity organizations are currently achieving great success in improving their overall software engineering processes.**
- **Use the process characteristics and effects I have presented here to give you a **general** indication of what level of maturity you are at, and what can be improved.**
- **Testing processes are not conducted in a vacuum. In general, assume the relative immaturity of the processes around you will prevent you from maturing beyond one level above the organization.**