

# ***Process Oriented Development***

---

**<http://lbgeeks.com/gitc/pmProc.php>**

**June 3, 2008**

- **What is a Process?**
- **Management Processes**
- **Development Processes**
- **Process Maturity**
- **Software Project Documentation**

# *What is a Process?*

---

- **Usually we think of a process as a job or task run by an operating system**
- **But that's only part of the definition**
- **At minimum a process is one or more of:**
  - **Inputs**
  - **Transformational activities**
  - **Outputs**
- **Work alone does not a process make!**

# ***Management Processes***

---

- **Accounting: budget, estimate, report**
- **Human resources: recruit, hire, fire**
- **Procurement: analyze, contract, buy**
- **Planning: develop, execute, control**

# ***Development Processes***

---

- **Requirements: assess, specify, analyze**
- **Design: preliminary, detailed**
- **Implementation: code, develop**
- **Test: plan, script, execute**

# ***Process Maturity***

---

- **Process level representing quality:**
  - **0: Incomplete**
  - **1: Performed**
  - **2: Managed**
  - **3: Defined**
  - **4: Quantatatively Managed**
  - **5: Optimizing**
- **Levels 0-1 offer temporary benefit**
- **Levels 3-5 have permanent benefit**

# **Levels 0-1: Incomplete, Performed**

- **Incomplete:**
  - **Process may not exist at all**
  - **Or exist in part, like just activities**
- **Performed:**
  - **Process complete but informal**
  - **May vary from person to person**
  - **Not documented**
  - **Can be lost over time**

## ***Level 2: Managed***

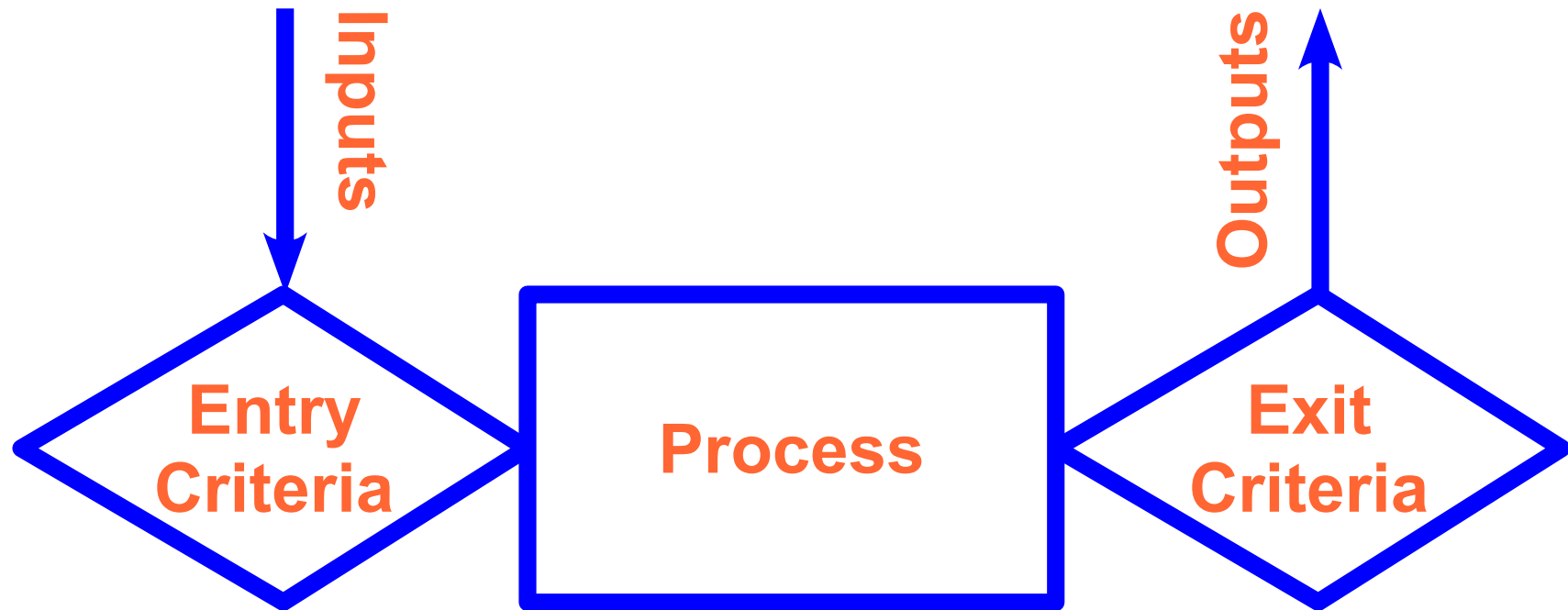
---



- **All team members use same process**
- **Process is documented**
- **New team members learn process after hire**

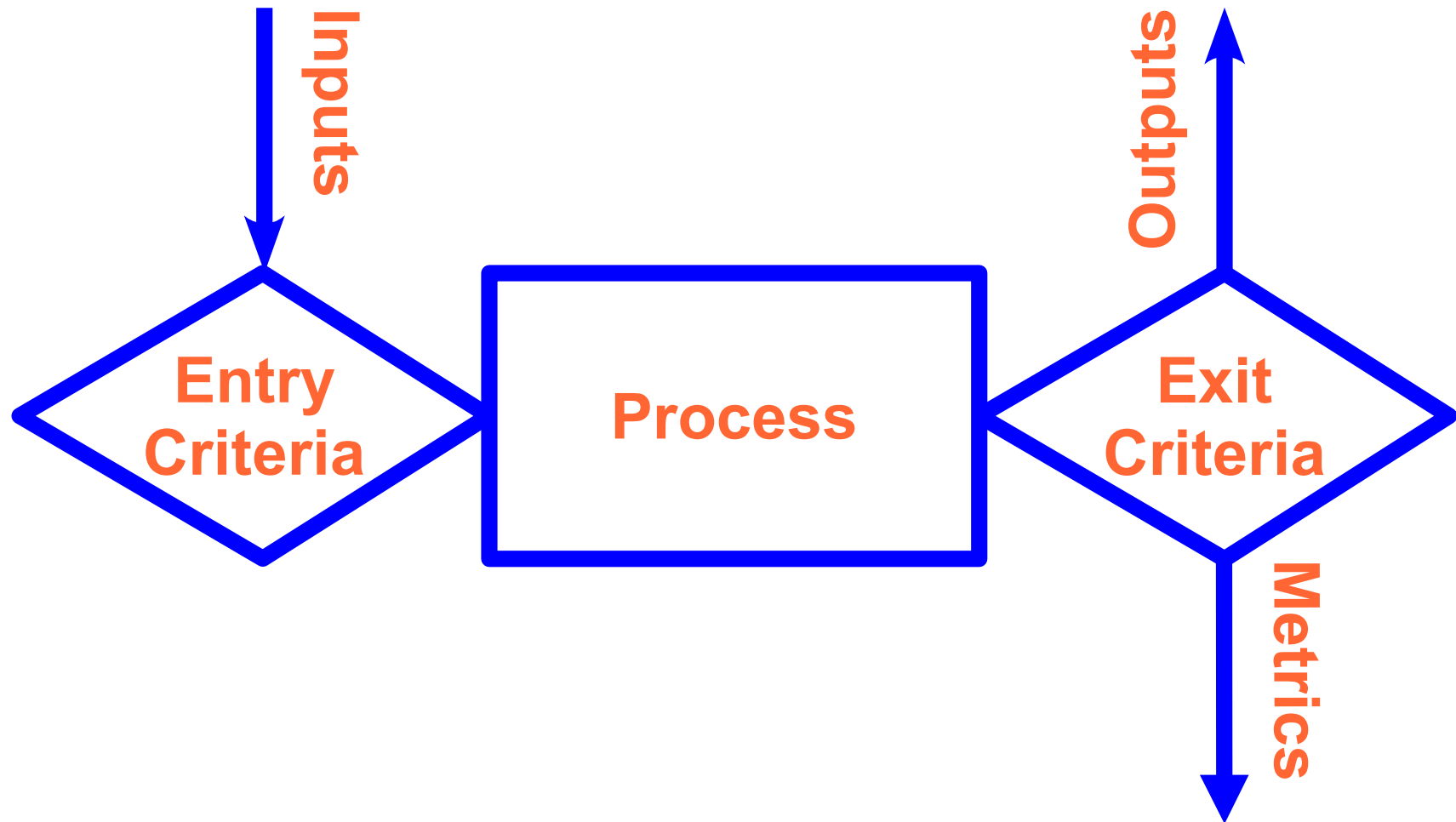
# ***Level 3: Defined***

---



- **Documentation far more rigorous**
- **Entry, exit, evaluation criteria known**
- **Consistent in application**

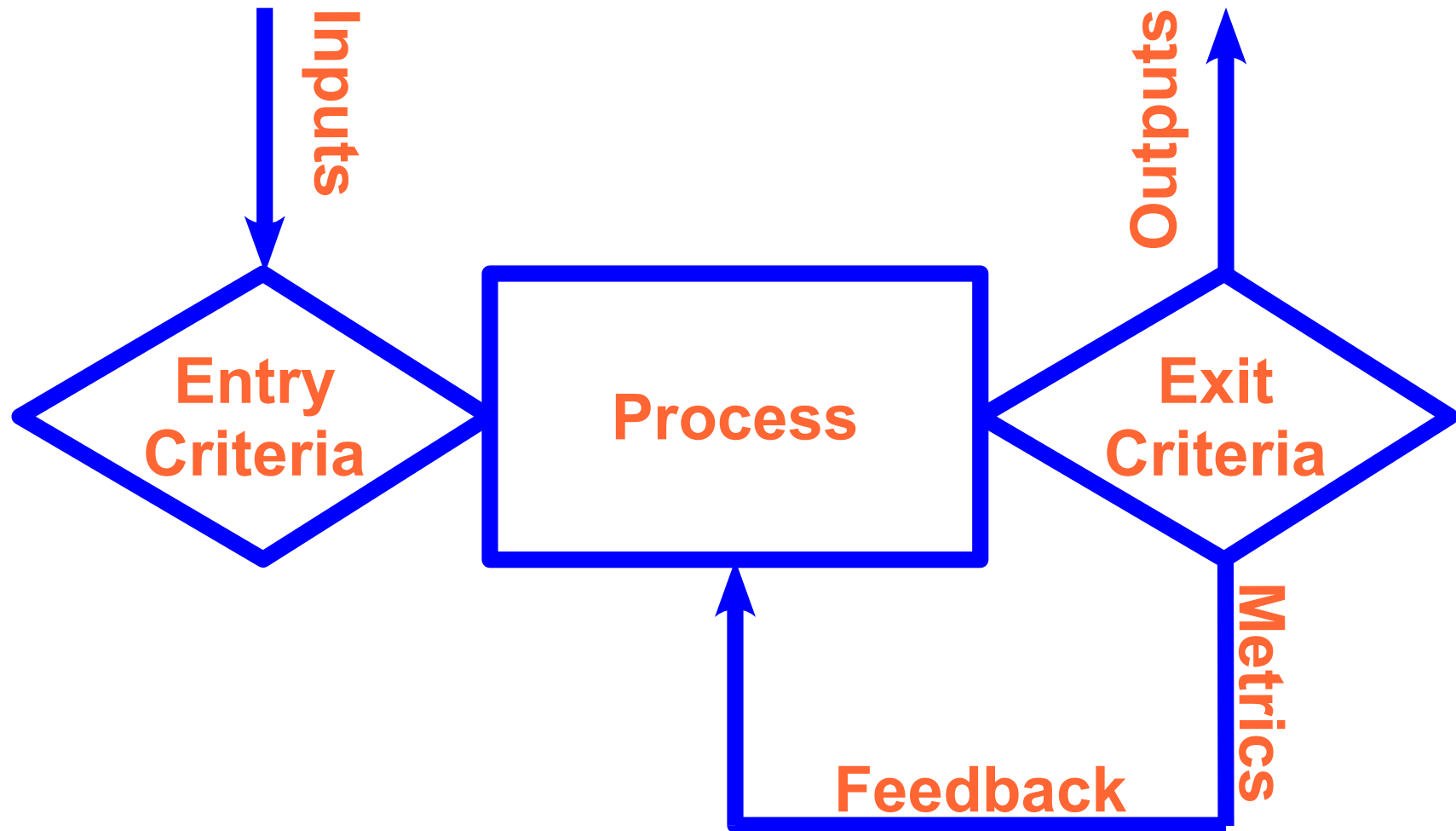
# Level 4: Quantitatively Managed



- **Process output is statistically measured**

# ***Level 5: Optimizing***

---



- **Process uses measurements to improve**

# ***Software Project Documentation***

---

- **Why is documentation so important?**
- **Technical:**
  - **System Requirements Specification**
  - **Software Requirements Specification**
  - **Software Design Description**
  - **Software Test Documentation**
- **Managerial:**
  - **Software Project Management Plan**
  - **Software Configuration Management Plan**
  - **Software Quality Assurance Plan**
  - **Software Validation and Verification Plan**

# *Why is Documentation so Important?*

- **Fun experiment: how much can one's mind remember?**
- **Enables:**
  - **Parallel development**
  - **Team co-ordination**
- **Reduces dependency on one person or specific team**
- **Allows completion of larger systems**
- **Reduces cost of fixing errors**
- **Increases quality**

# ***Requirements Specification***

---

- **Contents:**
  - **Needs assessment**
  - **Business requirements**
  - **Functional requirements**
  - **Non-functional requirements**
  - **Traceability matrix root**
- **Division from system into hardware and software after allocation process**
- **Notion of how customer will use product (software or system)**

# ***Software Design Description***

---

- **How will software implement requirements?**
- **Preliminary:**
  - **Object models**
  - **Sequence diagrams**
  - **Across objects**
- **Detailed:**
  - **Table design**
  - **Interface design and prototype**
  - **Code analysis (pseudocode)**
  - **Within objects**

# **Software Test Documentation (1 of 2)**

- **Requirements:**
  - **Against testing, a parallel activity**
  - **Constraints on budget, schedule**
  - **Standards compliance**
- **Plans:**
  - **How will system requirements be tested?**
  - **Assigns requirement to test case**
  - **Analysis, experiment, inspection, verification methods**

# **Software Test Documentation (2 of 2)**

- **Procedures:**
  - Specified process (IPO) for each case
  - Entry, exit criteria where necessary
  - Pass/fail criteria for evaluation
- **Steps:**
  - Procedure for executing test
  - Written without ambiguity
  - Allow novice to complete, judge test

# *Summary*

---

- **Process has specific definition**
- **Various levels: 0 (incomplete) through 5 (optimizing)**
- **Higher level impls:**
  - **Better definition, documentation**
  - **More positive impact on project quality**
  - **Permanence across groups, projects**
- **Software documentation is critical:**
  - **Helps with memory, organization**
  - **Allows parallel development**
  - **Reduces cost and risk**