How to Write a
for YOUR COMPANY

1. Introduction

MicroTools is proposing to assist YOUR COMPANY in improving the existing software process. The purpose of this project is to both improve the process already in place, but also better position YOUR COMPANY for obtaining future contracts. The deliverable from this project would be a Software Process Procedures and Policy Manual tailored to YOUR COMPANY's unique corporate environment, history and existing policies.

The general way this Manual is used is that it will define the "full-blown" software process at YOUR COMPANY. Each new software project will create a software project plan that will tailor these requirements to the specific requirements of the project. For example, an R&D project may forego some of the formal reviews and testing. A specific contract may have different documentation requirements.

As part of this project, MicroTools will perform the following basic activities:

- Assess the Existing Process
- Create a Proposed Plan
- Review the Proposed Plan with YOUR COMPANY
- Implement the Plan

2. Assessment of the Existing Process

MicroTools will initially assess the state of the entire software development process at YOUR COMPANY. This will be accomplished through document review and working closely with YOUR COMPANY personnel. At the end of this phase, a report will be issued and the results will be reviewed with YOUR COMPANY personnel.

The process will be evaluated against the 5 levels of Software Process Maturity as defined by the Software Engineering Institute. The major aspects levels 2-4 will be assessed. (There are no processes in level 1 and level 5 is beyond the scope of this proposal). The following breakdown lists questions that will be addressed at each level.

2.1. SEI Maturity Model Level II

2.1.1. Life Cycle Assessment
- What are the current life cycle phases (Planning, Requirements, Design, Implementation, Test, Verification, etc) of most of the software projects at YOUR COMPANY?
R&D Projects
Contracted Projects

- Define the tasks that take place during each of these phases
- Define the organization responsibilities for these tasks

2.1.2. **Project Planning**

- What planning occurs currently when a project is started?
- How are resources (financial, capital, and human) allocated?
- Organizationally, how are software projects staffed?
- What is the connection between software staffing and other disciplines?
  (how many hats do the software engineers wear?)
- What tools are used for project planning?
- How is project planning handled throughout the life cycle?

2.1.3. **Requirements Management**

- How are requirements identified?
- Who is responsible for partitioning system / product requirements into hardware / software requirements?
- How are changes in requirements tracked?
- What tools are used for requirements management?
- How is requirements trace-ability handled?
- How is requirements management handled throughout the life cycle?

2.1.4. **Project Tracking**

- What milestones are tracked during a project?
- How are these milestones measured?
- Who is responsible for project tracking?
- What metrics are accumulated through project tracking?

2.1.5. **Configuration Management**

- What tools are used to manage configurations?
- What items are identified to come under configuration management? (i.e. documents, software modules, make files, executables, etc).
- What is the identification method (naming conventions) used for each item's configuration (probably different for each type of item)?
- What levels of control are placed on each item?
- What mechanisms are in place to control changes?
- Do all documents have a version/revision level?
- Is there a common place for all software documents?

2.1.6. **Intellectual Property Protection**

- How are inventions identified during the development?
- How is trade secret material handled?
- What safeguards are in place to protect intellectual property from escaping?
- Who is responsible for protecting the intellectual property at YOUR COMPANY?

2.1.7. **Backup Procedures**

- What material is backed up? (Documents, source code, tools, etc)
• Where are the backups stored?
• How are the backups identified?
• How often is the backup procedure tested?

2.2. SEI Maturity Model Level III

2.2.1. Software Development Process

2.2.1.1. Tool Selection
• What are the existing development tools in use at YOUR COMPANY?
• How are tool costs assessed during the proposal or during R&D projects?
• Who is responsible for version control of the tools?

2.2.1.2. Development Methodology
• What are the current software development methodologies in use at YOUR COMPANY? (Waterfall, Functional Decomposition, Piece-wise refinement, Rapid Prototyping, Agile, Extreme Programming)
• Who is responsible for choosing a development methodology?

2.2.1.3. Documentation Requirements
• Which of the following documents are part of the current software process?
  • Software Development Plan
  • Software Test Plan
  • Systems Requirements Document
  • Interface Control Document
  • Software Requirements Specification
  • Detailed Design Document
  • Software Test Procedure
  • Release Notes
  • Version Description Document
• What other documents do you use during normal software development?

2.2.1.4. Scheduling
• How are schedules developed during the software development process?
• How often are schedules updated?
• Who is responsible for schedules during development?

2.2.1.5. Resource Allocation
• How are resources allocated during development?
• Who is responsible for allocating resources?

2.2.1.6. Subcontractor Management (if applicable)
• How are subcontractors managed during software development?
• How is quality of performance monitored during development?

2.2.1.7. Coding Standards
• What coding standards are in place?
• Are there company wide naming conventions?
• Does the code from all developers look the same?

2.2.1.8. **Test Requirements**

2.2.1.8.1. Levels of Testing / Formal and Informal
• What are the different levels of testing required during development? (Module or Unit Testing, Integration Testing, Design Verification, Beta Testing, Field Trials, etc)
• Which of the tests are formal and which are informal?

2.2.1.8.2. Resources and Tools
• Are the tools used in formal and informal testing under any kind of configuration control?
• When are the resources and tools identified during the software development process?
• How is the principle of independent verification implemented?
• Which test can be done by the software programmer and which must be done by a different person?

2.2.1.8.3. Make Versus Buy of the test environment
• When are decisions made concerning Make vs Buy of the resources identified as necessary for testing?
• Who is responsible for making this decision?

2.2.1.8.4. Regression Testing Policies
• Is there a policy that defines when a failure requires various levels of regression testing?
• Can small changes be made without completely re-testing a function?
• Who is responsible for making those decisions?

2.2.1.9. **Reviews - Customer and Internal**
• Which of the following reviews are part of the normal software development?
  - Requirements Reviews
  - Preliminary Design Reviews
  - Critical Design Reviews
  - Code Reviews
  - Test Readiness Reviews
• Is there a document that defines what happens in each of those reviews?
• Which reviews are done with the customer and which are internal only?
• Who is in attendance at these reviews?
• How are minutes and action items recorded?
• Who maintains and tracks action items at all reviews?

2.2.1.10. **Problem Tracking During Development**

2.2.1.10.1. Tools
• What tools are used for tracking problems found during development?

2.2.1.10.2. Responsibility / Oversight
• Who is responsible for tracking problems during development?
• Who is responsible to decide if a problem is fixed, only documented or otherwise disposed of?

2.2.11. Risk Management
• What formal or informal risk analyses are performed on the software?
• Who is responsible for determining the need for risk assessment?
• How is risk tracked throughout the process?
• How does risk analyses feed forward into the scheduling?

2.2.12. Software Safety
• Who is responsible for identifying safety issues in the software?
• Who is responsible for determining the safety of the software?

2.2. Software Maintenance Process

2.2.2.1. Change Management
• Are there any changes to the development change control during the maintenance phase?
• Who is responsible for determining whether a problem requires a new release?
• Is there a change control board to determine what changes are to go into what release? Who does decide that?

2.2.2.2. Regression Testing Requirements
• Is there a policy that defines unique maintenance requirements when a failure requires various levels of regression testing?
• Can small changes be made during maintenance without completely re-testing a function.
• Who is responsible for making those decisions?

2.2.2.3. Resource Management
• Does the same team that does software development also perform software maintenance?
• Do you have any legacy software that requires maintenance but for whom you have no in-house expertise?
• How are the development tools used during development maintained to allow maintenance?
• Can the development environment for all supported software products be re-created?

2.2.2.4. Warranty
• What is your policy on the cost to your customers of maintenance repairs?
• How long is free phone support provided?
• Who decides whether or not a change is a bug fix or a feature?

2.2.2.5. Subcontractor Management
• Are there any special tools or software used by subcontractors that are not owned by YOUR COMPANY?
2.2.3. **Software Quality Assurance**
- Is there a separate organization responsible for quality control at YOUR COMPANY?
- Is the head of the SQA report at the same level or above the Engineering department head?
- Is there a Software Quality Assurance organization?
- What is the responsibility of the SQA organization?

2.2.3.1. **Software Quality Plan**
- Does YOUR COMPANY have a Software Quality Plan?
- Who is responsible for implementing this plan

2.2.3.2. **ISO 9000 Considerations**
- Is YOUR COMPANY ISO 9000 certified?
- When will you be re-certified?
- Were ISO 9000-3 software requirements addressed during the initial certification?

2.3. **SEI Maturity Model Level IV**

2.3.1. **Productivity Metrics**
- What metrics does YOUR COMPANY have to measure individual software engineers productivity?
- What metrics does YOUR COMPANY have to measure software project productivity?
- How often are these metrics re-calibrated?

2.3.2. **Process Metrics**
- What metrics does YOUR COMPANY have to measure the quality of the Software Process?
- How often are these metrics re-calibrated?

2.3.3. **Product Quality Metrics**
- What metrics does YOUR COMPANY have to measure the quality of the Software products? (Bugs / line of code or Bugs / Function Point?)
- How often are these metrics re-calibrated?

3. **Create a Proposed Plan**

Working from the results of the assessment, MicroTools will propose a plan to document the software process at YOUR COMPANY. This plan will include a complete outline (Table of Contents) of the Software Process Procedures and Policy Manual.
4. Review the Proposed Plan with YOUR COMPANY

This plan will be reviewed with YOUR COMPANY and decisions made concerning what parts of the plan should be implemented. Many of these policies and procedures will merely be documenting the existing processes. However, there will be some that will have a cost and schedule impact on new projects. During this phase, these will be carefully reviewed based on their estimated cost impact.

5. Implement the Plan

Using the existing process documents, the IEEE standards, and information obtained during the assessment and reviews, MicroTools will implement the Software Process Procedures and Policy Manual?

6. Review the Procedures Manual

Once the Procedures Manual is put in place, we will review this with YOUR COMPANY and make any suggested changes.

---

The following are the 5 Levels as defined by the SEI.

**Level 1** - Characterized by chaos, periodic panics, and heroic efforts required by individuals to successfully complete projects. Few if any processes in place; successes may not be repeatable.

**Level 2** - Software project tracking, requirements management, realistic planning, and configuration management processes are in place; successful practices can be repeated.

**Level 3** - Standard software development and maintenance processes are integrated throughout an organization; a Software Engineering Process Group is in place to oversee software processes, and training programs are used to ensure understanding and compliance.

**Level 4** - Metrics are used to track productivity, processes, and products. Project performance is predictable, and quality is consistently high.

**Level 5** - The focus is on continuous process improvement. The impact of new processes and technologies can be predicted and effectively implemented when required.