APPENDIX F. PROCESS MATURITY

F.1 What is a Process?

Teamwork and processes get projects accomplished. A short definition of "process" is [11]:

> *A set of activities performed for a given purpose.*

A more insightful definition is [60]:

> *The logical organization of people, materials, energy, equipment, and procedures into work activities designed to produce a specified end result. A system in operation to produce an output of higher value than that of the sum of its parts.*

A small number of key processes must be in place for CVISN development and deployment. Key technical processes are described in the CVISN technical guides. Key project management processes are described in the guide you are holding and in the *CVISN Guide to Phase Planning and Tracking* [44]. Since software is such a significant element of any ITS project, your state might be interested in assessing the maturity of its own software acquisition process [11].

F.2 What is Process Maturity?

Process maturity is what gives you confidence when you visit the ophthalmologist's office for laser vision correction, or when you take out-of-town guests to your favorite restaurant. The success of those organizations isn't by luck on a good day. Rather, it is built upon **repeatable robust processes that work in spite of fallible human beings**.

Process maturity has been modeled as stages though which organizations evolve as they define, implement, measure, control, and improve their processes. In an effort to permanently transform software development from esoteric wizardry into predictable engineering, the Software Engineering Institute at Carnegie-Mellon University in 1988 first published the seminal concept of a process maturity framework, best known as the software development process capability maturity model [61]. See Figure F-1.

This framework has proven to be so profoundly useful that it has been applied to several other areas such as system engineering, and most recently the *Software Acquisition Capability Maturity Model* [11]. As a project manager you have the power to exert firm control over many of your project's processes. Awareness of process maturity helps you envision what is possible; smooth-running processes (even for "simple" tasks like status reporting) are more efficient, effective, and satisfying.

F.3 Process Maturity Framework

The Software Engineering Institute's landmark process maturity model framework has five levels:

Level 1. Initial

Often better described by what it lacks rather than by what it has. There are no formalized procedures, no cost estimates, no project plans, no change controls. It is marked by a lot of crisis behavior. "Initial" may be a polite word for "chaotic".

Level 2. Repeatable

Key process areas are recognized. There is requirements management, project planning, project tracking and oversight, subcontract management, quality assurance, and configuration management. Can't do better than this, right?

Level 3. Defined

Whereas the key process areas in Level 2 are recognized, they may vary greatly in implementation from project to project or with personality style. In Level 3 they are institutionalized; there is an organizational process focus; and as a result processes are defined and written down. For example, there will be an organizational Cost Estimating Manual, and a Software Quality Assurance Plan. There will be training programs. There will be inter-group coordination, and peer-level reviews. All projects use an approved, tailored version of the organizations' standard processes.

Level 4. Managed

At this level not only is product development managed in a projectized environment, but the processes themselves are also managed, and in a quantitative manner. For example, the Cost Estimating Manual would be updated periodically with feedback on actual costs such that parametric estimates could be made based on experience. The rate of a product's problem failure reports (presumably declining) would be measured and projected into the future as a measure of product quality. Processes are quantitatively understood and controlled.

Level 5. Optimizing *(note the "ing" implying continual improvement)*

Continual process improvement is enabled by quantitative feedback from each process and by piloting innovative ideas and technologies. For example there is a defect prevention process, not just a defect cure process. There is a process in place to inject new technology into development, such as new computer-aided design tools or computer-aided software engineering tools. There is a process in place for changing processes.

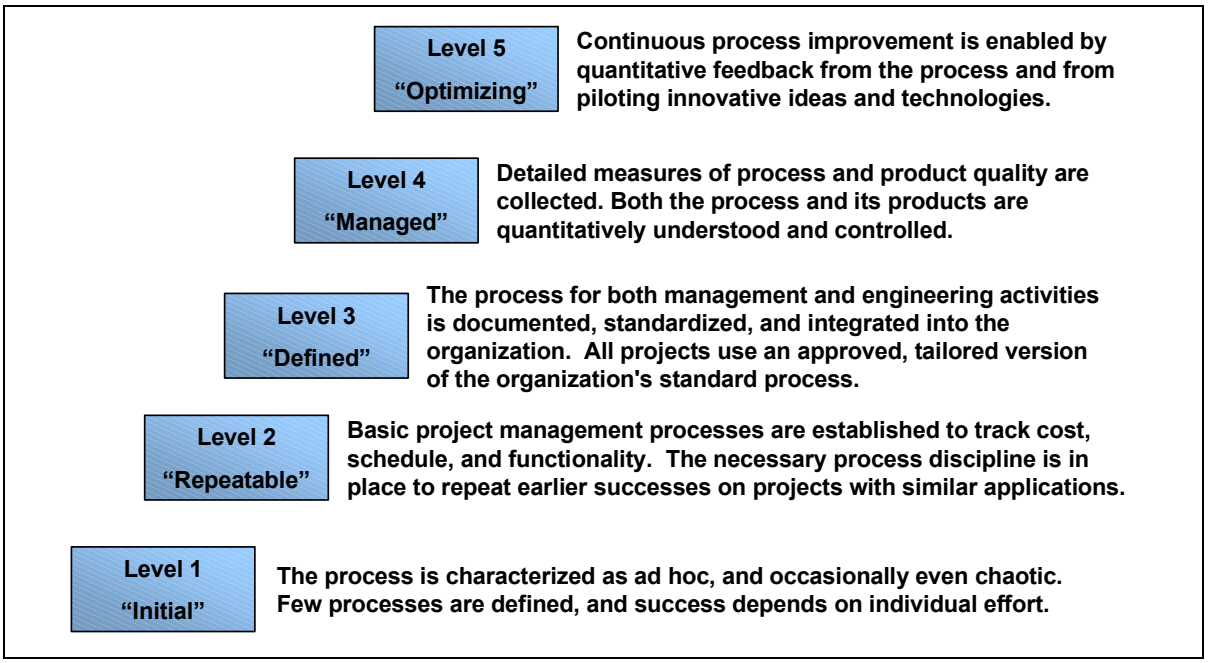

Figure F-1. Process Capability Maturity Model

Even after all this description you still might wonder what these levels would feel like in ordinary experience. Figure F-2 is an analogy.

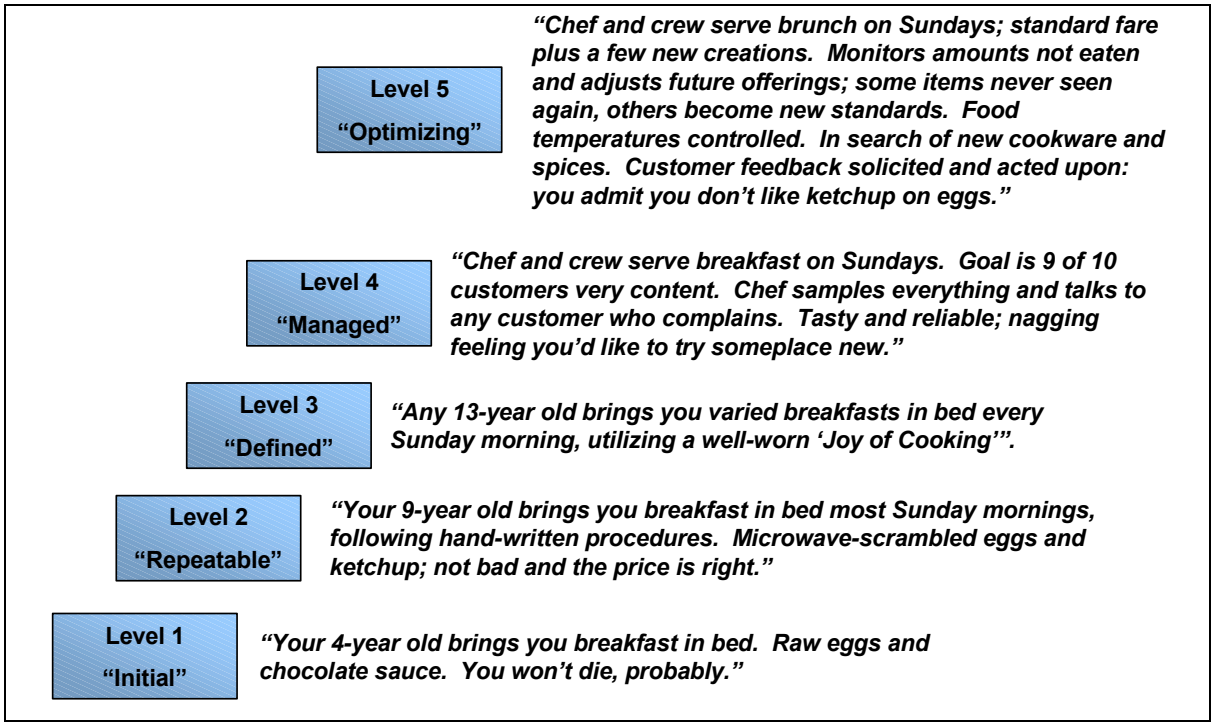

Figure F-2. Analogy of Process Capability Maturity Levels

Of particular interest to the states will be the *Software Acquisition Capability Maturity Model* [11], whose major activities are shown in Figure F-3.

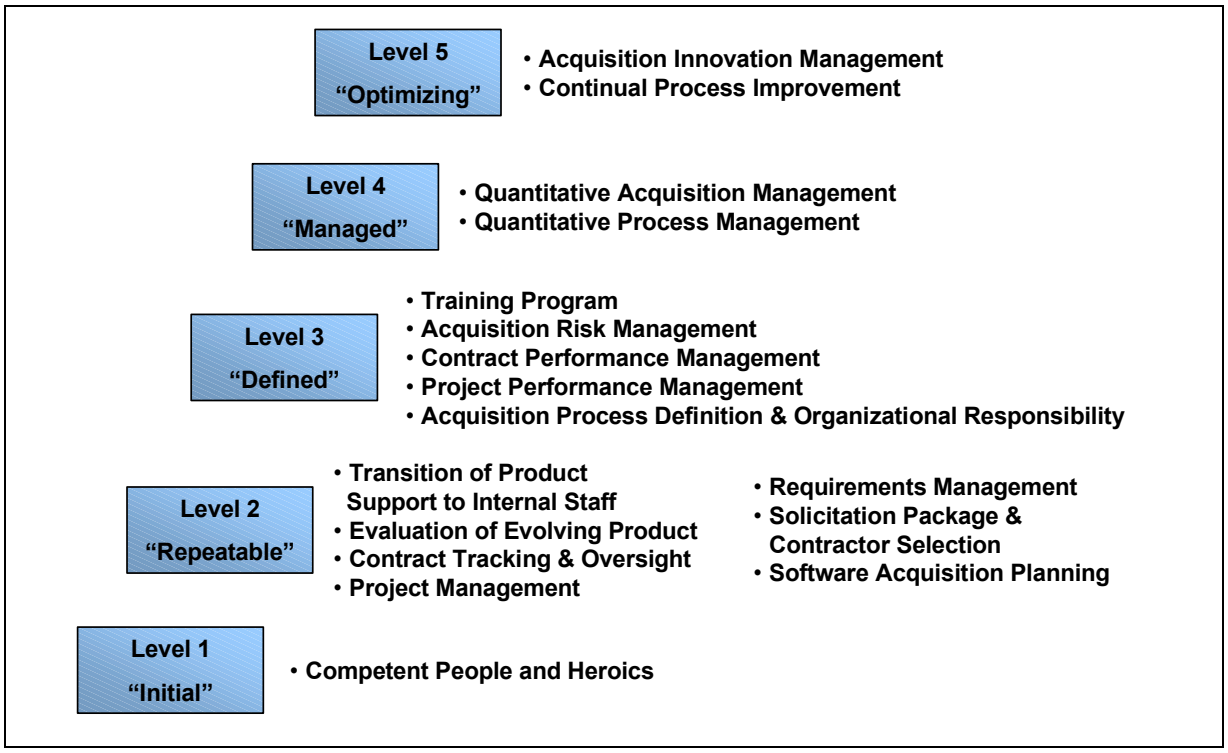

Figure F-3. Software Acquisition Capability Maturity Levels