F - 1: Process Infrastructure

Good ways of working are invaluable. They should be protected, maintained and improved. To do this a software development organization should put in place the necessary infrastructure. A process infrastructure provides the organizational structures, usually described on the organization's organization chart, defines the necessary roles together with their terms of reference and the lines of communication. It also describes the relationships between the roles that will provide good process support.

This infrastructure is critical for support of software development processes and ways of working, perhaps more critical than for the maintenance of other organizational assets, because of the intangible nature of software development. Without a sustaining infrastructure ways of working can be eroded, neglected over time, or damaged by misguided management.
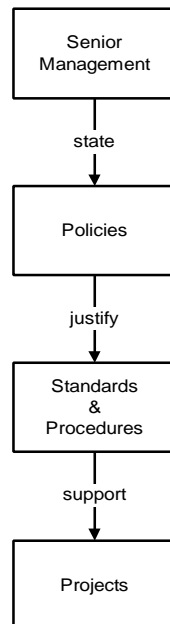
Fortunately an effective process infrastructure is simple and can be documented concisely, describing how the various players involved in the support and development of good working practices will work together. It can also be used to define the scope of the organization's process assets, and distinguish these from the staff's professional skills.

The generic model, described below, can provide the basis for an organizational process infrastructure:

The primary group in the infrastructure are the software developers. They perform the working practices and use the process assets. This group contains the organization's technical 'know how', experience and practical knowledge of the working practices. To aid them in their work they will use a variety of models, and product and process standards. Some may be documented and explicit, others undocumented but familiar and valued 'custom and practice'.

The next group is the senior management group. This group is responsible for commissioning software development work and ultimately responsible for the performance of the software produced. As such this group will express clearly their view, not only what should be produced but what processes should be in place to produce it. That is, the senior management group, as policy makers, should express their will on the manner in which software development work is conducted for them. This is most effectively done as a set of software development policies – senior managers are policy makers.

The relationship between the working practices of software developers and the views of the senior management are mediated by the senior management's policies, that say *what* should be done, and the software developer's processes and procedures that describe *how* the work is to be done. The policies direct and justify the processes and procedures; in effect the policies are the requirements to be satisfied by the processes and procedures.

```
        ┌─────────────────┐
        │     Senior      │
        │   Management    │
        └─────────────────┘
                 │
               state
                 ▼
        ┌─────────────────┐
        │    Policies     │
        └─────────────────┘
                 │
              justify
                 ▼
        ┌─────────────────┐
        │    Standards    │
        │        &        │
        │    Procedures   │
        └─────────────────┘
                 │
              support
                 ▼
        ┌─────────────────┐
        │    Projects     │
        └─────────────────┘
```

Policy (or its equivalent) is critical. In the absence of clear direction from senior management on ways of working they may be changed in any manner over time as technical staff, tools and technology change. Conversely, senior management attempting to set standards and establish working practices directly can also be problematic; they lack the current, in-depth of knowledge necessary to specify working practices.

The relationship:

*senior management>***make***>policies>***justify***>processes<***use***<software developers*

 is necessary but not in itself sufficient to direct software development practice. Actual practice needs to be made visible and assessed, and deficiencies made good where there are found. This requires two further groups or roles.

The first of these is the Software Quality Assurance (SQA) group. This group's function is to aid software developers in their planning and use of the working practices, and to monitor their use and report this to the senior management group. In effect this group acts as the eyes of senior

management. Given this potentially delicate situation clear well communicated terms of reference for the SQA group are essential if it is to function. These may state:

- Why the organization has an SQA group.
- Who the SQA group acts for, and reports to.
- The scope of working practice within the scope of the SQA group's activities.
- The obligations of the SQA to development groups.
- The obligations of the development groups to the SQA group.
- The means for escalating issues should they arise.

The effectiveness of the SQA depends critically on the response of the senior management group to the operation of the SQA group, as perceived by the rest of the organization.
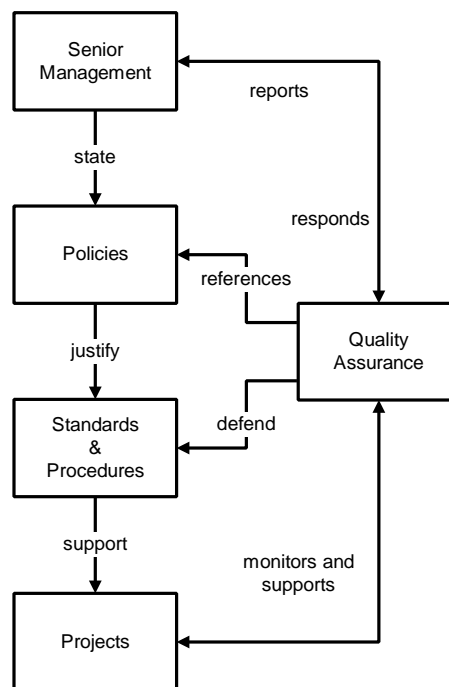

Senior management should use, and be seen to use, the SQA function to inform themselves on software development practice and its conformance to the 'will' of senior management. The nature of this 'will' varies from organization to organization. In some organizations senior managers will have clear views and direct ways of working, in other simple consensus and consistency of working practices among developers alone is seen as desirable. In either case it is important that the senior managers care about ways of working and maintain control of them.

This is demonstrated by the process infrastructure. In organizations without well-defined ways of working the establishment of policy, and monitoring of conformance to policy by an SQA

f1

function is enough to begin the development of valued process assets, simply by cutting though the debate on conflicting opinions and ways of working. Policy directs ways of working:
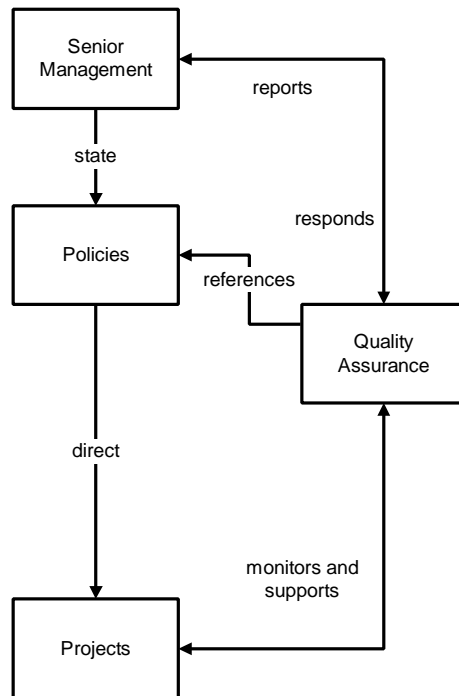
Whereas processes and procedures unmanaged or directed by senior management can rapidly deteriorate as SQA looses its rationale.


4
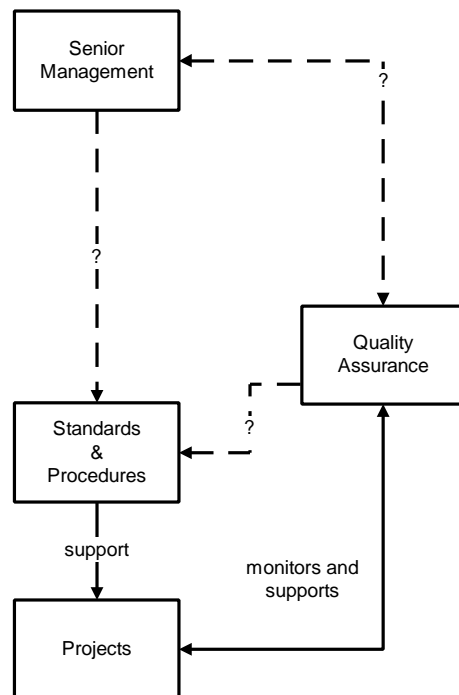
The second group is the Software Process Group (sometimes known as the Software Engineering Process Group or SEPG – however the term engineering can be inappropriate or unwelcome in some organizations). This group's concern is the improved operation and development of working processes. Where the SQA group defends the status quo – monitoring what is happening and comparing this with existing standards and procedures the process group looks for more effective ways of working within the scope of the senior management's software development policies

Like the SQA group the process group should have terms of reference to scope its activities. These may include:

- The objective of the process group to actively and measurably improve the performance of the organization's software development activities.
- Qualifications for group members
- The size and structure of the group
- The group's interactions with software managers and staff, for example how it will access information and request work from software development managers and staff
- The group's process improvement and control practices.

Ideally these two groups are distinct, containing different individuals (although a representative from the SQA function may be included to provided a view of the SQA functions). The two groups can be considered as the organization's 'process executive' and 'process legislature'. In smaller organizations maintaining two distinct groups is may impose too much of an overhead and be combined into a single group. In such cases the different functions should be carefully distinguished to avoid confusion.
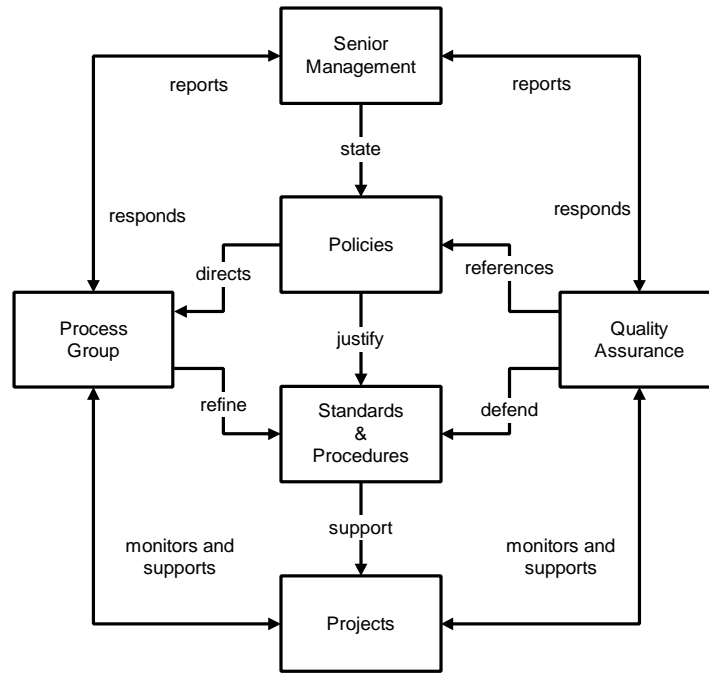
The process assets – working practices and the supporting standards and procedures - that this process infrastructure supports and develops will contain a number of different types of items:

- A software development lifecycle – to provide a framework for development plans and a context for procedures and guidelines.
- Procedures and guidelines to direct and support activities.
- Product standards.
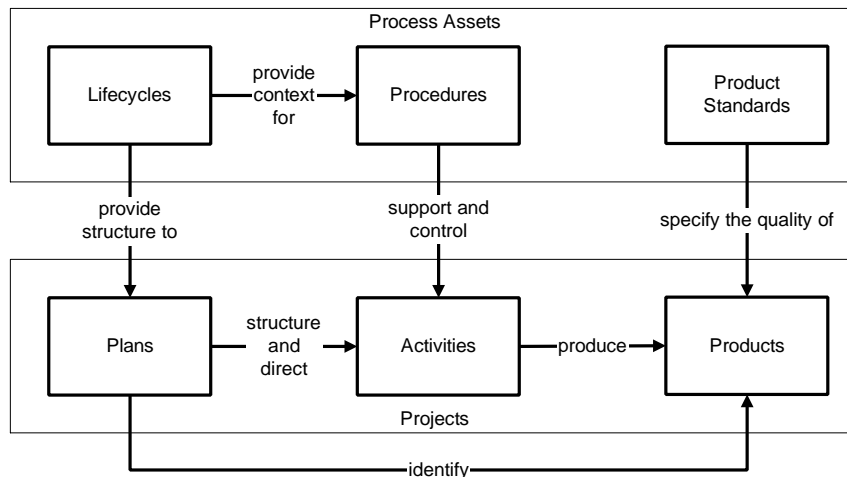
In the early stages a software development lifecycle is the most valuable asset. It contains a series of phases and the products produced in each phase, and is used to provide a template

f1

for software development plans. With increasing sophistication roles and responsibilities become well defined and processes, procedures and standards will develop.