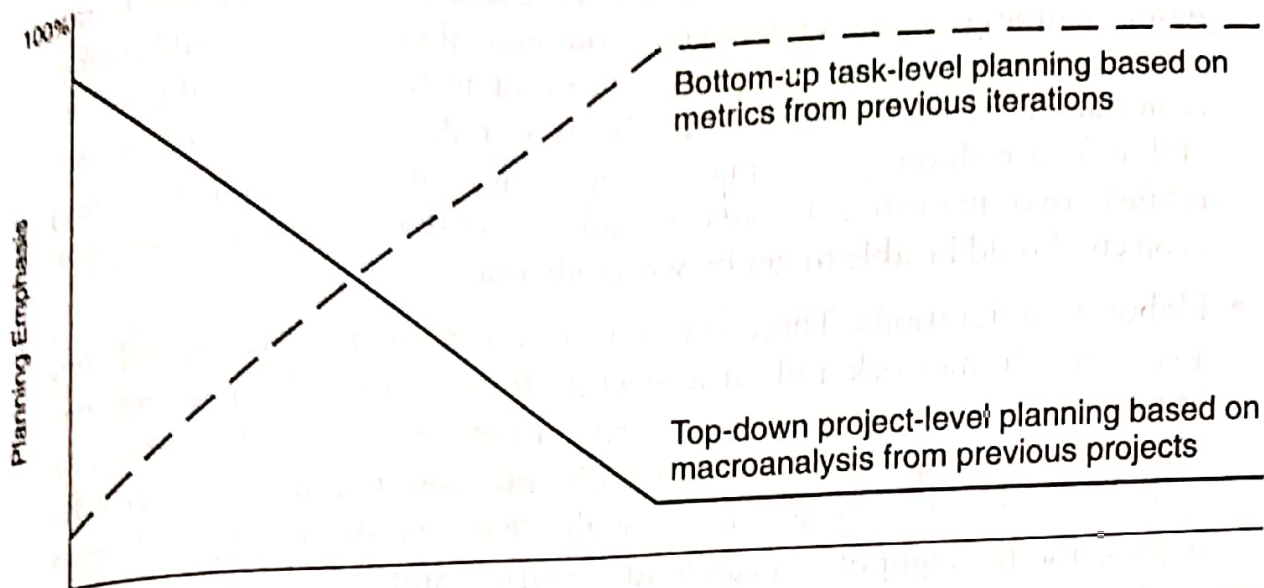


10.4 THE ITERATION PLANNING PROCESS 18.3.20 (MRA1)

So far, this discussion has dealt only with the application-independent aspects of budgeting and scheduling. Another dimension of planning is concerned with defining the actual sequence of intermediate results. Planning the content and schedule of the major milestones and their intermediate iterations is probably the most tangible form of the overall risk management plan. An evolutionary build plan is important because there are always adjustments in build content and schedule as early conjecture evolves into well-understood project circumstances.



Engineering Stage		Production Stage	
Inception	Elaboration	Construction	Transition

Feasibility iterations Architecture iterations Usable iterations Product releases

Engineering stage planning emphasis:

- Macro-level task estimation for production-stage artifacts
- Micro-level task estimation for engineering artifacts
- Stakeholder concurrence
- Coarse-grained variance analysis of actual vs. planned expenditures
- Tuning the top-down project-independent planning guidelines into project-specific planning guidelines
- WBS definition and elaboration

Production stage planning emphasis:

- Micro-level task estimation for production-stage artifacts
- Macro-level task estimation for maintenance of engineering artifacts
- Stakeholder concurrence
- Fine-grained variance analysis of actual vs. planned expenditures

FIGURE 10-4. Planning balance throughout the life cycle

A generic build progression and general guidelines on the number of iterations in each phase are described next. *Iteration* is used to mean a complete synchronization across the project, with a well-orchestrated global assessment of the entire project baseline. Other micro-iterations, such as monthly, weekly, or daily builds, are performed en route to these project-level synchronization points.

- Inception iterations. The early prototyping activities integrate the foundation components of a candidate architecture and provide an executable

framework for elaborating the critical use cases of the system. The framework includes existing components, commercial components, and custom prototypes sufficient to demonstrate a candidate architecture and sufficient requirements understanding to establish a credible business case, vision, and software development plan. Large scale, custom developments may require two iterations to achieve an acceptable prototype, but most projects should be able to get by with only one.

- **Elaboration iterations.** These iterations result in an architecture, including a complete framework and infrastructure for execution. Upon completion of the architecture iteration, a few critical use cases should be demonstrable: (1) initializing the architecture, (2) injecting a scenario to drive the worst case data processing flow through the system (for example, the peak transaction throughput or peak load scenario), and (3) injecting a scenario to drive the worst case control flow through the system (for example, orchestrating the fault tolerance use cases). Most projects should plan on two iterations to achieve an acceptable architecture baseline. Unprecedented architectures may require additional iterations, whereas projects built on a well established architecture framework can probably get by with a single iteration.

- **Construction iterations.** Most projects require at least two major construction iterations: an alpha release and a beta release. An alpha release would include executable capability for all the critical use cases. It usually represents only about 70% of the total product breadth and performs at quality levels (performance and reliability) below those expected in the final product. A beta release typically provides 95% of the total product capability breadth and achieves some of the important quality attributes. Typically, however, a few more features need to be completed, and improvements in robustness and performance are necessary for the final product release to be acceptable. Although most projects need at least two construction iterations, there are many reasons to add one or two more in order to manage risks or optimize resource expenditures.

- **Transition iterations.** Most projects use a single iteration to transition a beta release into the final product. Again, numerous informal, small-scale iterations may be necessary to resolve all the defects, incorporate beta feedback, and incorporate performance improvements. However, because of the overhead associated with a full-scale transition to the user community, most projects learn to live with a single iteration between a beta release and the final product release.

The general guideline is that most projects will use between four and nine iterations. The typical project would have the following six-iteration profile:

- One iteration in inception: an architecture prototype
- Two iterations in elaboration: architecture prototype and architecture baseline
- Two iterations in construction: alpha and beta releases
- One iteration in transition: product release

Highly precededented projects with a predefined architecture, or very small-scale projects, could get away with a single iteration in a combined inception and elaboration phase and could produce a product efficiently with the overhead of only four iterations. A very large or unprecedented project with many stakeholders may require an additional inception iteration and two additional iterations in construction, for a total of nine iterations. The resulting management overhead may be well worth the cost to ensure proper risk management and stakeholder synchronization.