Time Box Scheduling

Technology is not the strongest driver of productivity for a software project.

If Technology Isn’t, Then what is?

♦ The length of a project is the single most important factor influencing productivity.
♦ Short projects with smaller teams are more productive than longer projects with larger teams.
♦ This is a “simple” concept, but it is also complex since time boxing requires:
  – Commitment from all participants
  – Rethinking the traditional management engagement process.
  – Strict control over the inclusion of requirements in each cycle.
Components of Time Box Scheduling

♦ User Involvement
♦ Senior Management Agreement
♦ Business Objectives
♦ Project Empowerment
♦ Scheduling
♦ Backing Out Changes
♦ Testing

User Involvement

♦ Users must agree with the approach and be involved through at all times.
♦ Users must acknowledge that priorities must be set and observed.
♦ This is a departure from the normal “analysis and design” processes.
♦ Schedule comes first, features come second.
♦ Priorities for managing “requirements in the box”
  – Essential requirements
  – Required but could be in the future
  – Include only if effort is available
  – Specifically not included
**Senior Management Agreement**

- Senior management must agree on the time box approach
- They must resist any attempts to break the time box
- With only a few exceptions, 6 months is a maximum duration for a time box project
- Staying with the time box is a matter of resolve by all the participants

**Business Objectives**

- Focus on key business objectives
  - Critical success factors can be used to define these objectives
  - Iterative deliverables can verify that value is being produced
- “Business needs” are the goal, not how they are met
  - Delivery of value needs to be controlled by architecture
  - If the delivered component fits the architecture, its internals are subject to local optimization
- Nothing requires a specific technology or toolset, provided the requirements are focused on the business needs
**Project Empowerment**

- The team must be allowed to make decisions in order to maintain the schedule
- Waiting for decisions doesn't work, PERIOD
- Project sponsor must be available at all times. They must be:
  - Capable
  - Involved
  - Responsible
  - Decisive
- Make decisions within the framework of the business objectives, supported by the architecture

**Scheduling**

- It’s easier to estimate how much work can be done in a fixed amount of time, than how much time it will take to do a fixed amount of work
- Time boxing restricts “scope creep”
- The focus on deadlines allows resources to be added at the end and still maintain the schedule
- Experience says “time lost can NEVER be made up”
**Backing Out Changes**

- Features that are not feasible or break the time box must be removed
- Good configuration management is essential
  - Undoing a feature should be a matter of regressing the configuration
  - Continuous testing (test first development) allows regressions to take place in a seamless manner

**Testing**

- Continuous integration and testing is essential for staying on schedule
- In phased-based system only unit tests can be performed before the system is assembled
  - These interfaces tests form the basis of time boxing the delivery.
  - The system runs the unit tests at all times
Win / Win

♦ Core Requirements
♦ Commitment
♦ Resources

Core Requirements

♦ *Ensures that only key requirements are developed*
  - The duration is fixed
  - The features can come or go
♦ “Nice to haves” are deferred to the next release cycle.
Commitment

- *Over time enthusiasm diminishes*
  - Staying on schedule requires constant vigilance
- *Once important items are forgotten all is lost*
  - Continuous testing against essential deliverables
- *Six month time boxing maximizes enthusiasm*

Resources

- *Finishing on time, on budget removes the backlog for the next phase*
- *Pressure on the next phase is removed in the beginning of this phase*