Management Tracking and Control of Cost, Schedule

BASELINES — "stakes in the ground" that set expectations for next stage in project
- Allocated – set of stable requirements upon which design based
- Developmental – stable design upon which implementation based
- Products – stable delivered product upon which maintenance based

INSTABILITY — changes to baselines ... manager’s nightmare
- Rework — reversal of decision made solid in baseline
- Relative cost to fix is exponential with distance to affected baseline
- Tell-tale sign => TBDs in baselines.

RE-ESTIMATION — required when baseline changes.

PROGRESS — usually measured against a predefined PLAN.

MANAGEMENT TRACKING — basic methods
- Establish plan (expectations) for stage of project – must be in terms of specific measurable or countable things
- Measure the things called for in plan ... "actuals"
- Compare ACTUALS against PLANNED
- Give results in the form of VARIANCE REPORTS, where

\[ VAR = PLANNED - ACTUAL, \text{ or } \%VAR = 100 \times \frac{VAR}{ACTUAL} \]

MANAGEMENT CONTROL — basic approach
- Gather status information
- Project undesirable impacts
- Make management decision to control undesirable impacts
- Re-estimate and re-plan for next stage
TRACKING & CONTROL OCCUR AT MANY LEVELS

Program level — major external (to customer) milestones
- Expectations likely to be fixed early on, hard to change
- Low visibility into what's "really" going on
- Focus on $ spent, external milestones (delivery dates).
- Replanned only when "critical" problems are made visible.

Phase by phase — internal, software specific
- Planned allocation of effort and schedule across phases
- Internal products whose completion indicate progress
- Completion criteria for each phase ... is this step a reasonable foundation (baseline) for next step?
- Replanned when progress or cost indicators raise RED FLAG ...
  – specific task taking much longer than planned
  – number of "undecided" (TBD) items unacceptably high
- Detailed plan developed for next stage based on "current" project status (i.e., information from just the completed phase).

Task level, within phases — detailed
- Helpful to benchmark first of many similar tasks, to refine estimates.
- May be needed for high-risk tasks.
- More common for coding/unit testing phases where lots of small, well-defined tasks defined.
- Simplest completion criteria, since granularity so low.
The QUALITY PROBLEM – Definitions

- conformance to requirements (IBM)
- conformance to explicitly documented development standards
- conformance to implicit characteristics that are expected of all professionally developed software ("-ilities")

How is Quality Measured?

- During formal testing ... when testers detect lack of conformance to requirements (via Discrepancy Reports).
- After the fact ... when customers complain of lack of conformance to requirements (via User Error Reports).
- During development ... results of reviews, amount of rework

A pragmatic metric: DEFECTS / KSLOC

- Can apply at any point, against current estimated/actual size
- Better one: LATENT DEFECTS / KSLOC ... hard to know!

What Quality Assessment methods are used?

- Formal Reviews (see textbook section 17.3 for guidelines and checklists for such reviews.)
- Informal "peer" reviews — walkthroughs and inspections
- Quality Audits — conformance to standards, requirements traceability
- Software Testing

Role of Standards

- establishes common "look and feel" of products => facilitates review
- Adherence to standards /= correctness => just checking standards is insufficient.
How can Quality be Managed?

- Establish Philosophy with respect to defects: where will effort be spent — on prevention or on detection?
- Planning — set quality goals (but how?), define means of tracking
  - Need historical database for projecting likely number and kinds of defects
  - Define “quality” for each intermediate product (standards, technical ”goodness”)
  - Select the quality assessment methods, allocate sufficient resources (for testing, reviews, inspections/walkthroughs)
- Development — execute the plan and track results
  - Pass all reviews
  - Strict adherence to standards
  - Try R–E–A–L HARD!

Published Results on Inspections (Glen W. Russell, Bell-Northern Research, "Experience with Inspection on Ultralarge-Scale Developments"

- Focus was code inspections
- There is an optimal rate of inspection (organization/project specific): for them, 150 SLOC/hr
  - 0.4 major defects/hr, or 2.5 hrs/major defect, ~1 defect/hr.
- Cost justification: 4.5 labor days to fix product error vs 2.5 hrs to detect and fix => ~ 30 hrs/major defect!
- #Intrinsic defect density is 50–100 defects/KSLOC
- Defect removal efficiency: estimated to be 40–75%.
- Inspections can be up to 20X more efficient than testing
- Inspections particularly good at finding MISSING/EXTRA errors.

MANAGEMENT TRACKING

- Establish plan (expectations) for defect removal per stage of project, and plot actuals vs planned. Pay close attention to the quality of inspections — their goal is to find defects.