Principles to Action:

Effective Technical Program Management
Layers of Expertise and Responsibility

Richard M. Bixler
netboyz@comcast.net

updated: 17 September 2020

- Layering of **Principles and Actions**, **Structure and Plan**, and **Tooling**
  - Principles/Governance include Advocacy and Aggregation, which are Driven by Product/Program Structure and Plan; themselves Driven using Structured Data with Tools.
  - **Principles**, **Governance**, **Advocacy**, **Aggregation**
    - **Principles and Governance**
      - ***Make it your mission as PM to meet business goal components*** *of Net Present Value (NPV): Schedule, Product Cost, Program Cost*
        - Plan and execute both a successful Product concept and design, and successful Program logistics plan and execution. Success depends on both product success and logistical success.
        - Nobody will have depth of knowledge of the project like an active Program Manager. This fact alone is your principal leverage within the org, and with all levels of management.
        - Everybody else will have specialized knowledge, a few with knowledge of adjacent functions; but none will have the overall grasp that the PM will attain.
      - ***PM: Be or Become Excellent*** *at something needed in EACH project-contributing organization, plus in the overall management of the program*
        - **"People respect you more if they get the truth as opposed to a bunch of fluff" – Marcus Lemonis**
        - Start with quantification of the project by product element.
        - Incorporate organization structure (as users with identified uses of equipment). Expand program plan to cross-functional/cross-organization dependencies and deliverables
        - Expand further to use the plan for active management of deliverables plus active management of product changes, and mitigation of issues encountered
        - Org workers love to talk about their project work. Get them to educate you; and use that knowledge in building and managing the project plan.
        - PM should NOT take the role of functional expert of an organizational component, even if PM has that expertise. Organization should retain that responsibility, and PM may provide opinions.
    - **Advocacy** and **Aggregation**
      - ***Program Overwatch****: PM must know the project from top to bottom, and rides shotgun to ensure success on all critical factors and tasks. As necessary, initiates interventions, escalations, provisioning, meetings or whatever is necessary to keep critical tasks executing to plan.*

- ***Bureaucracy:*** *The PM should be the enemy of bureaucracy. On the one hand, due diligence and good practice must be performed on all tasks. Don't get the product out and then crash and burn. But bureaucracy expressed as delay for the sake of doing it slowly should be addressed by the PM, if it is not avoidable in the first place. Bureaucratic delay should never be allowed to affect a critical path.*
- ***Connection****: EE, ME, FW, SW, NPI, Marketing, Management: Continuous connection with mutual respect and credibility with program Principals.*
- ***Critical Path Intervention****: Critical path defines the shortest time to program plan completion. This task sequence requires primary attention by the PM, with intervention as needed to keep on track. Critical Path undoubtedly runs through major integration tasks, which are excellent target activities to keep the organization focused on achieving.*

- ***Integration****: Organizations often focus and organize around developing aspects of their central technologies. In my experience, often there is nobody identified for realization and integration of the resulting artifacts. Organizing, planning and provisioning Integration therefore becomes de-facto a problem to be addressed by Program Management. Problems to be solved include identification of use and configurations, builds, tool scripting, IT integration, duration and timing, finance, and cross-function dependencies and execution.*
- ***Roadblock Removal and Escalation****: Keep the critical and near-critical paths on track.*
- ***Standing Meetings****, Communication: Maintain communication with all levels of the program. These are the mechanisms to see roadblocks or lagging progress, to initiate intervention, to see progress. Project elements, integrations, management connection, senior management connection. Senior Management connection is often well handled by a quarterly status review that includes/embeds a functional demo.*
- ***Track, Update, Structure, Recovery, New****: As the Program evolves, and the PM must modify and update the Program Plans for any of these events, and then propagate knowledge of the updates through organizations that will change operation to implement the change.*
- ***Reporting****: To Management at all levels using Summary and Backup Detail as required or anticipated, for coordinated management of all Orgs with involvement in the Program.*
- ***Advocacy****: To Management, Orgs, Partners, and Employees across the Company, at all levels, using Summary and Backup Detail as required and permitted. Drives and aggregates program information for formal reviews, probably creates sections on Allocation to major tasks, BOM structure, Schedule integration, and critical material.*
- *Focus on these listed topics is visible and necessary but presentation focus is insufficient by itself. Effectiveness is significantly enhanced by supporting Structure and Plan tasks described in the section just below.*

- Program Management Organizational Leadership
  - ***"If you don't know where you are going, any road will get you there" – Lewis Carroll***

- ***Planning*** *results in realistic goals and how-to's throughout the organizational hierarchy, which is a critical component of Leadership. "Realistic" is due to due diligence or deeper representation of dynamics and overwhelms planning based on hope.*
  - Plan for success. Develop aggressive (but not foolhardy) plans keeping error recovery off the critical path. Everything won't fail at once. Most tasks are going to succeed. You want a clear picture of the Critical Path, not containing embedded slack that causes misidentification of that path. Tasks not on the critical path inherently contain slack. A problem on the critical path will be mitigated to some degree, likely taking advantage of slack from previously not-critical tasks. So don't bury slack in task durations scheduled. The PM should allocate slack in the program via planned task sequencing and task dependencies.
  - People and orgs are good at working toward tangible objectives. But creating such objectives is a special talent of Leaders. Rather than starting a planning meeting with a blank sheet, it's better for the PM to start with at least a "strawman" plan that can be criticized, modified, embellished, accepted or rejected by participants. That may have been developed by the PM, or by a core group. Groups of people are better at comment and critique than at synthesis, especially for plans extending beyond their areas of direct responsibility. PM collects info and input, but is generally the one who has to knit it all back together even when everybody leaves the discussion thinking everything is fine.
  - Updating/mitigating/re-planning/updating plan hierarchy keeps goals realistic.
  - Planning and management based on realistic plan described here constitutes real operational leadership, and that is what you will become respected for.
- *What can be actually accomplished: Quantitative Management significantly improves project operation and outcomes*
  - Quantification, and management by quantification, gives PM a truly significant grasp of program, respect among workers, and (ultimately) respect of management.
  - Significant quantification requires tooling and structure/planning discipline
  - There is opportunity here but nobody, and no organization, is stepping up.
- *One More Principle to remember at all times:*
  - ***"You're not the first one to have this problem." – me***
  - Come up with a concise description of your problem and look it up on the Internet. Refine the description until you start getting results in the ballpark. Learn from others. Many have written up their solutions. Use good knowledgebases: Google Assistant is very good at interpreting your query and finding answers. For coding issues, StackOverflow, Microsoft TechCommunity.

- Structure and Plan
  - *"If you don't know your numbers, you don't know your business" – Marcus Lemonis*
  - Quantification links Principles/Governance: Program Principles to Plan > Execution > Tracking / Update Cycle
    - *Build vision/plan for full project, full duration*
      - Plan should be **Detailed enough to describe best current understanding of full effort but not so detailed as to be brittle**. Plan should be at a level "necessary to describe task, deliverables, and dependencies for major efforts, with due diligence". Plan should always be organization's Best Current Understanding of the effort, not a Hope Plan or a worst-case plan. However, current understanding may include estimates and tentative plans if those are best current understanding. Due diligence should replace those estimates as rapidly as possible: any Hope Plan should not remain best current estimate for long.
      - **Plan should work at all levels** of program hierarchy.
      - **Keep it up to date**: Best Current Understanding of Product structure, mitigation, changes, uncertainties/plan alternatives
      - **It may take several iterations** to develop a plan that addresses the business NPV needs: Schedule, Product Cost, Program Cost. Iterate until confident that you have the best possible plan; not the first available plan.
  - **It is executed by developing a structured/Hierarchical Plan at all levels and managing to it, updating it as conditions are actualized, better understood, or change**: Vision / Plan / Quantification / Execution / Tracking / Mitigation / Re-Planning
    - *The Structure Example described in this article illustrates hierarchical participants to be planned/executed. You would determine analogous structure for subset efforts, or for other disciplines.*
    - *Example in this article is illustrated using **Power OpI** (Power Operational Intelligence) processes and tools:*

      https://www.softtoyssoftware.com/

      https://www.youtube.com/channel/UCj_c0djARr1-eICgUbfbCbg/featured
    - ***PM creates initial plan** based on product structure, organizational components, and product lifecycle, then **validates/negotiates** with stakeholder orgs **within overall program structure**; then **make reality match Vision** and Plan as updated. **Update plan** continuously (at least weekly) to incorporate changes and mitigations.*
    - ***Greatest leverage: Plan, Drive, Track, Manage critical path to Integration Points**. Mitigate issues as needed to keep the Integration Point dates.*
    - ***Schedule drives Logistics; and Logistics Drive Finance**.*
    - *Quantification of Schedule, logistics (materials - goods and services acquisition, usage and timing of materials), and Finance is the core of Quantitative Management, as applied to Program Management of a New Product Development Project. The Tools and Consulting described in the banners at the top of this page address these topics.*

- *In most organizations, the PM has better summary and detail view of the Program than any other individual or institution. Therefore, top-level responsibility for these Quantitative Management tasks, and often for detailed implementation of them, falls to the Program Manager. This responsibility may be informal, but it is real. At program completion, or failure, the PM is the First to Go, or to be Ignored. "One Neck to Grab". To really be effective, the Program Manager must be highly conversant on the following topics:*
  - Envision Structure of Complex Project and Quantify it in a Tangible and Useful Plan. The Program Manager is responsible for guiding, and knitting together, views of planning and execution for requirements, design, operations, and organizational operation. The program Structure may include, or even be based principally upon, principles of Agile methodology, Critical Path, or Waterfall methodology ideally as described in the Waterfall and Agile article on the Power OpI website.
    - Best Practice: PM guides quantified program plan development by identifying, with help and validation from program Principals, the major integration points defining the top-level structure of the program from end-to-end; and building the quantified plan around them; and then working down to increasingly detailed development prior to integration points, and testing following from them.
    - PM identifies elements, tasks, and dependencies within and across all organizations participating in the project, and quantitatively knits them together to create a quantified and comprehensive Program Plan. Initial start by PM, the reviewed/updated with stakeholder orgs within overall program structure.
    - The quantified plan is kept up to date as events and changes occur over the duration of the program.
    - PM makes quantitative Predictions and determines where and how to place bets, to insure them, and how to initiate recovery if needed.
  - Elements of Quantitative Management
    - "Schedule": Task Sequence, Duration, Dependencies and Deliverables, resulting in program Dates. Build the schedule first, based on development-lifecycle principles (phases, sprints, lead times etc.).
      - *EE, ME, and FW Board, Module, Assembly, Integration, Systems Builds and Test, by development Phase (development and bringup; initial qual test, formal qual and Compliance test, new product introduction, customer pilot and FCS).*
      - *Plan the Multiple levels of Integration, Dependencies and Deliverables for sub-assembly builds, element integration, and system integration.*
      - *Product Software: Product block function/file organization, Function across product block structure, Structured sprint plan, feature organization into Feature Blocks (migrated function, new function, desirable/optional/follow-up), with Integration Points for function/management/kernel/middleware/application*
      - *Sprints sequenced to meet dependencies and deliverables among SW and HW entities, with sync points among software and hardware elements.*

- *DevOps, Private branch and Integration Branch management, SQA, and internal distribution cycle, and product Release*
- *Manufacturing Operations: Supply Chain, Prototype builds/CM, Manufacturing BOM structure, Manufacturing Test.*
- *HW Test sequencing: Bringup P0, Pre-Qual P1, Qual P2, Compliance PP*
  - *Test Plans: Function, corners, stress. Bringup/function, performance, coverage, yield, verification/validation, qualification, compliance, pre- and formal certification, internal/external end user*

- Materials: Allocation, BOM, Logistics Data
  - *Based on the Schedule, build the Logistics/Materials plan next supporting the schedule.*
  - *Plan Allocation of Built Items by Organization, by identified Use and Configuration within the Program Sequence*
  - *Plan the supporting Build Plan, multi-level, bringing up Product and also Operations capability to build the product. Plan items to build, and items to buy for integration. Plan product prototypes, and include test system configurations.*
  - *Plan details of Built and Purchased Items at all levels, tied to Build Plan - board components, subassembly components, system components, software components, licenses, incremental labor. Critical components, cables, power supplies, rack components, third-party plug-ins, test and library software, licenses.*
  - *For hardware elements specifically, encourage changes made during project execution to be backward-compatible at the lowest level possible. If updating one element requires update to another element as well, that multiplies update logistics by an order of magnitude. Make sure developers are aware of consequences of design updates.*
- Finance
  - *From Schedule and Logistics, incorporate financial detail: costs of materials, fabrication, services, licenses, labor etc. Identify manufacturers, distributors, proto cost, tooling, lead times, payment terms for each item.*
  - *Plan, review, manage and report Expense and Capital; Prototype material, NRE, Tooling, Contracting; Budget and Actual; by Calendar and Fiscal Quarter; by Org, Element, Phase and many other sorts. Short-notice Ad-Hoc Reporting as needed.*
- Iterate
  - *Iterate schedule, logistics and finance to meet as best possible, intended program schedule, product cost, and program cost.*

- *Quantified Management: what to plan*
  - Quantification enables and in fact forces you and the organizations to translate and align principles into action.

- Quantify Module config and use, Product Structure, Schedule, Material details. Quantify means detailed identification of elements, with numeric quantity, time, cost etc.
  - Quantify Schedule, Materials, and Finance. Coordinate methodologies among contributing organizations: Align HW phases with Sprint groupings, Ops lead times, integration time, test time. Plan for follow-up bugfix/function release.
  - Quantify Feature Block content, Sprints, DevOps, SQA, Release plans. Remember hardware is too expensive to rework in the field; plan full HW function at FCS that can be exploited by later SW feature releases.
  - Plan tasks for Development, Qual/Compliance/Certification, Rollout
  - JOIN quantified plan components to expand visibility and utility of the plan, and then Data-Mine JOINed plan for quantified/sequenced/dated Module/Org/Operational plans, Material plans for source/use, plans for Finance of program, fiscal year, each fiscal quarter.
- Structure project meetings
  - Virtualize across geographical and local venues
  - Set up and run meetings by project component, by assembly structure and Integration Points, by function (validation/verification/compliance/rollout/support/suppliers/external), by management levels (cross-functional, senior management). A project lead may be identified for each of these, or else the PM must lead them. If a project lead is identified, the PM must participate and remain current, to substitute as needed for the project lead, and to incorporate current events into the quantified program plan.

- *Building actionably-quantified Schedule, Materials and Finance plans is highly dependent on Tools described in the section following.*
- ***Schedule for Working Path First, to keep whole organization working at all times***
  - Favor planning based on "working path first" over a plan to put all-new functionality on everybody's critical path broadly in the program organization via prioritization of risky work over known work. That all-new work could hold up equipment for everybody. For example, people doing call-management software don't need the latest audio modules, they can use current-generation capability with little or no disadvantage. Provision as much of the org to proceed at all times, to introduce their functionality and to find and fix their problems before they themselves become critical to progress.
  - Technicality: Critical Path consists of tasks, leading to key milestones such as FCS, that have no slack. Slack is time over and above a task duration, that a task can slip without holding up a following, dependent task.
  - As much as possible, plan tasks, or task sequences, to execute in parallel particularly with un-compressible tasks such as builds, time-driven qual test, certification etc. Organizational management, not the PM, should be responsible to balance staffing availability to tasks.

- Manage critical path(s) with fervor. Identify bottlenecks and risks in advance and remove or mitigate them. Pay attention to critical path, near-critical paths, and paths that will become critical if any problem occurs (e.g. new fab cycle or certification would need to be inserted). Ahem: don't do anything un-safe, illegal, or that would be an unrealistic stretch.
- If a problem encountered on critical path can't be mitigated, preserve the critical path schedule by moving tasks to mitigate the problem OFF critical path, to a path in parallel to the critical path. Preserve the critical path schedule. Let organizations that are not directly dependent on the task or element with the problem, continue to find and resolve their own problems.
  - Never make any substantial part of the program wait for completion of a critical element from a program organizational component. Move that work off of everybody's critical path, until it's ready for integration.
  - Backfill the problem tasks with mitigation tasks until solution is ready to re-integrate
  - When ready, backfill the solution element into real dependent uses, not necessarily to all uses. No need to pay for distributed elements twice, only those fulfilling direct dependency.
- Some people will say a Plan can never be accurate. Nevertheless, it is necessary for financial/logistical planning, coordination; and when identifiable specifics are found inaccurate, for managing change. A managed plan will converge toward accuracy as the project progresses.
  - Agile processes are generally interpreted to defer decisions: to make decisions when specifics are not yet known could be characterized as bureaucratic or non-value added.

    The form of work described in this article contrasts with this, advocating building a plan based on information available including reasonable assumptions, in order to build a best-currently-known path; then re-plan and re-plan as information changes or becomes available, to maintain a best-currently-known path.

    Then predictions can be made based on that path that affect work sequence, equipment need, costs and revenue timing and so forth, as needed to move lead-time or spending off currently known critical paths.

    "If you don't know where you're going, any road will get you there" (Lewis Carroll), versus "Get Directions" on Google Maps showing constantly-updated optimal path to your endpoint from your current location, updating constantly on traffic, or on deviations that you may initiate. It boils down to whether you can identify and take an optimal path from your current position.
  - PM typically must integrate multiple dev methodologies anyway: HW phases; Agile SW feature development; DevOps/Branch management; manufacturing/partner disciplines et al.

  - Firmware and diagnostics generally schedule with board-level hardware not software, because of tight interdependencies with development and Manufacturing. Board-level mechanical parts schedule with board development
  - Just-in-time decision-making is more of a development methodology, than a program management methodology.
  - Not all areas of schedule are unknowable. Porting existing function, integrating existing functional library, incremental feature add are more know-able; than is development of new infrastructure/technology such as new transport, memory type etc.
  - Research organization's prior history for task-type – that is the best way to replace Hope with Due Diligence. If none, then define an "intended" duration. Refine plan to reality and keep history record for the next project.
  - Consider Central Limit Theorem: A large number of samples converges to a Gaussian distribution. If task history predicted-vs-actual duration data is available, it can be used to determine parameters that model that error population as a Gaussian distribution. That distribution function can provide predictability of a project based on similar task types. A Monte Carlo analysis using this distribution function with your project structure (see Crystal Ball software product), can be used to predict project performance at a stated level of confidence. My experience says this applies well to schedule, material, and financial planning.
  - If an org function "refuses" to plan, nevertheless it always happens that someone in the org structure is making an assumed plan on their behalf. That could be Finance, Marketing, senior management/business commitments. Best if org function (or PM) makes better-informed estimates.
  - Is schedule "un-knowable", or is lack of scheduling just avoidance of responsibility (to org coordination and finance)? Somebody in the org is taking responsibility for the schedule and its related finances and revenue. What is the value to a "best current understanding" approach, in order to realize that value? HW schedules have just as many risks as SW, plus more partners distributed geographically.
- Plan aggressive Intended Date and later Fallback Date for program completion (and follow-on). Work toward Intended Date, and don't lie or hide status. Project should NEVER fail due to lack of planning. PM and Management can't control physics but can control planning/execution/communication/coordination/outlook/mitigation.
- Escalate management of politics to highest level that can mitigate, for cases that affect program NPV – schedule, product cost, program cost.

- Tools to Quantify/Mine/Update/Report
  - PM Quantification
    - *Outlooks Manufacturing and DevOps planning, ahead by 6mo at least for finance, critical materials, assembly/CM planning, development dependency/sequencing. PM will be planning efforts 6-12 months ahead of engineering or manufacturing organizational attention to planning, and will be planning test equipment and builds in addition to product prototyping.*
    - *Capability fundamentals using Project, Excel, Access/SQL Server, Visio*
      - Excel Structured Tables, Index/Match-SUMIF, VBA, Pivot Tables from Structured Tables, and Pivot Tables from SQL QUERYs.
      - Linking Excel, Access, SQL Server, MS Project, Visio
      - Data Structure representations: Normalization, Allocation and use, BOM product structure by Phase, Schedule, Materials Detail
      - Structuring projects, System Development, Application and incorporation of Waterfall, Agile, multi-disciplines, Schedule Optimization
      - HTML, CSS and other Web tools, for communication of plans via the Program website/Sharepoint/Wiki.
      - Can't integrate cloud tools (Smartsheet, Google Sheets, MySQL, PostgreSQL et al) by PM; that would need significant dev support, rarely available for PM tools, so be careful on commitment to tools. Many cloud tools don't yet talk to each other without significant software development. It's a trap to sacrifice "functional integration" for "modern".
  - Formal Management/Release Tools
    - *Connecting Organizational operations tools, with PM tools. Ops tools usually constrained by procedure/labor for Operations organization formal release and for revenue/expense production, and thereby limit effectiveness for PM planning/tracking. PM plan needs to handle many entities for development, test and support that will never be released, are planned ahead of development and therefore use speculative structure, and which are needed throughout the program with instantaneous turnaround.*
    - *Manufacturing For product release, developed and use during the program for building protos and for bringup of Operations capabilities: MRP-based formal BOM; Testware/sequencing/procedures, assembly docs. For CM: integration, configuration, test; as product develops. Tools include Oracle, Arena, Corporate Finance. Determine a path to feed formal organizational operation tools from PM tools. Discourage Ops from re-inventing program-developed structure – that will negate experience and test gained on PM structures during development.*
    - *DevOps tools: Source Control/GitHub/Perforce, Bug Tracking/Bugzilla equivalents. Sprint tools/AI tracking/Jira et al. Dev plan should incorporate elements to be managed by these tools.*

- *Integrate/feed/report with PM-developed project tools*

- Program development Tracking and Coordination
  - *Hierarchical Project Meeting Structure includes the following forums*
    - Development/Working level by project component or major task (Compliance etc)
    - Integration/Logistics
    - Operational Management, Senior Management, Finance
    - Communication, updates, cheerleading across whole organization including program contributors and also bystander organizations.
    - Formal Governance reviews: Business Commit, Dev Commit, FCS Release Review
  - *Program Quantified plans will be a major component of ISO audit.*
  - ***Tools: Word** (agenda/minutes – working, tracking, planning), **Powerpoint** (group presentation, mgmt. summary)*
    - Project meeting agenda and tracking: For each forum, Cumulative accumulation of notes for each meeting instance: Status, Notes, Problems, Ais, then-current schedule points.
  - ***Jira**, **Basecamp**, **Asana** et al for action items*