Management By Objective

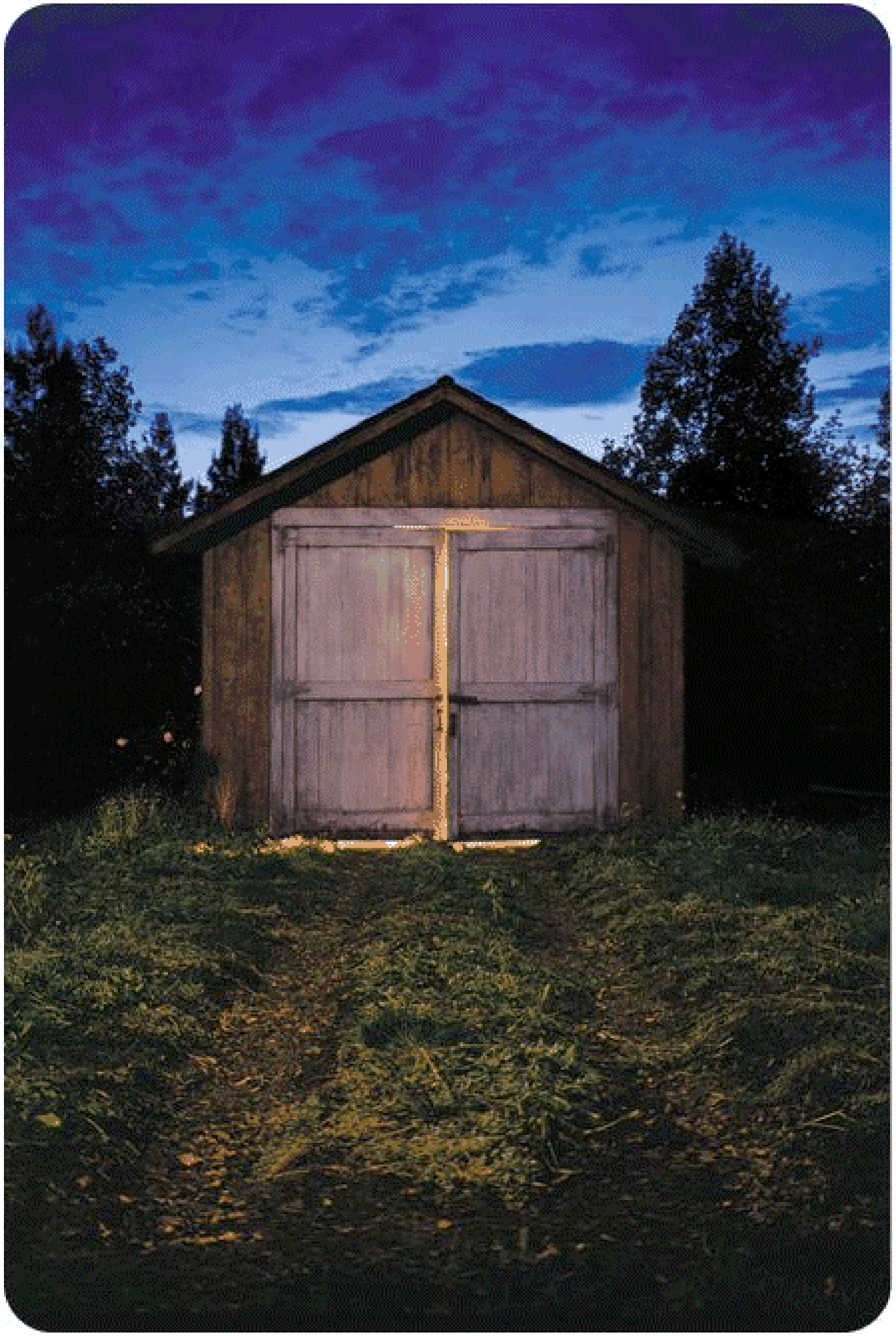
Mark DiPasquale

Hewlett-Packard Company

June 8, 2001

Agenda

- Introduction
- An MBO Model
- MBO Model Explained
- Templates
- Case Studies (brief samples)



Introduction

Objective

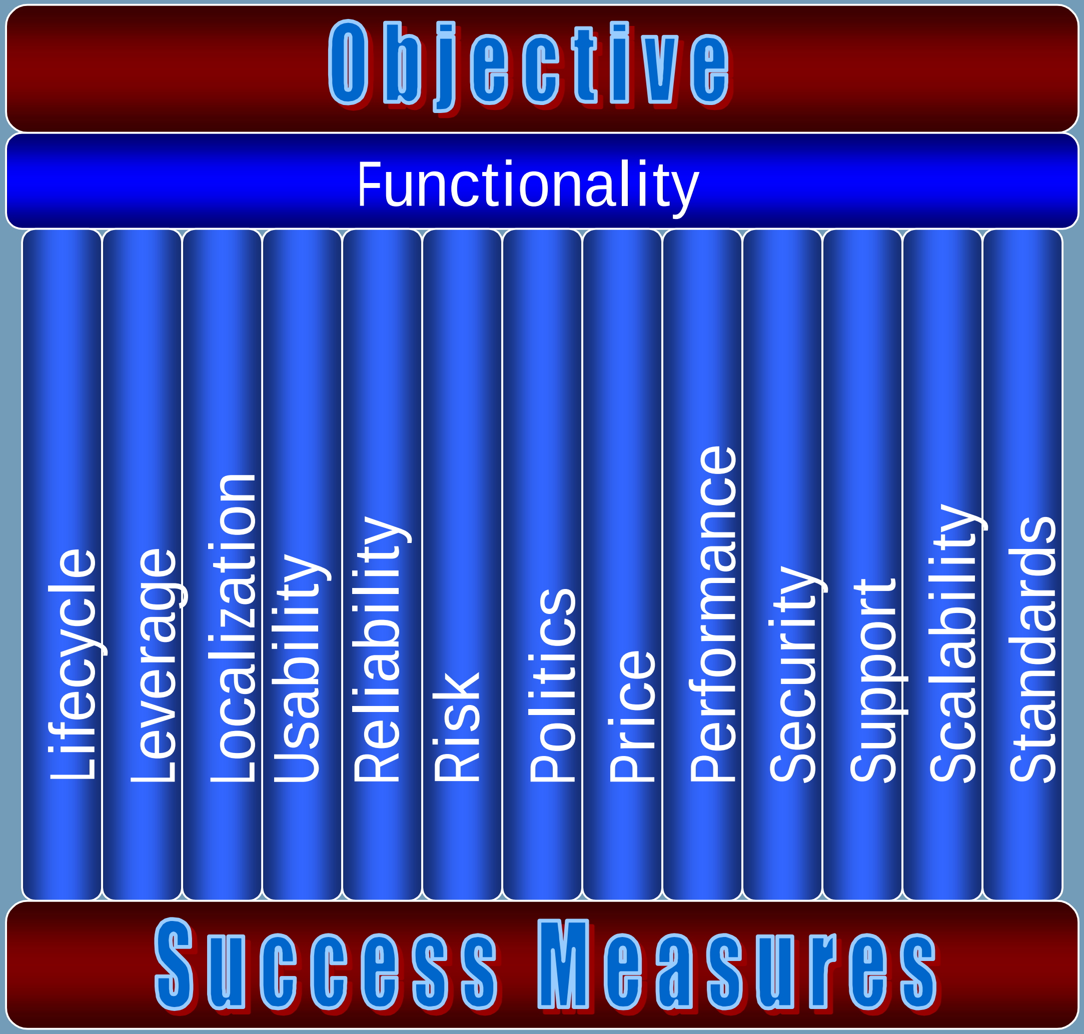
To provide a management by objective methodology for project management that is reasonably comprehensive and is always available.

- “by objective” – results focused
- “reasonably comprehensive” – covers most of the planning elements required for repeatable success. Guideline, not a panacea
- “always available” – can be *memorized*; minimal training and minimal lookup required

Presentation Objective

Leave this room able to to use 80% or more of what you learn in this presentation; 100% after a day’s study

The O’FL^{3}UR^{2}P^{3}S^5 Model



Objective

Functionality

Lifecycle

Leverage

Localization

Usability

Reliability

Risk

Politics

Price

Performance

Security

Support

Scalability

Standards

Success Measures

Objective

- “Begin with the end in mind.”
  - *“The 7 Habits of Highly Effective People,” Steven R. Covey*
  - Results-oriented
  - This philosophy helps one to ‘point the ship’ in the desired direction, to overcome obstacles, to gain consensus, and to stay on course.
- The objective defines the what, when, and measure of a project.
- Objectives are typically phrased with the word “To” at the beginning of the objective statement, as in “To what by when, measured how.” For example, a database vendor might have this business objective:

To be the number one database solutions vendor by January, 2003, as measured by world-wide market-share and profitability.

Functionality

- Divide & conquer
  - Identify all relevant deliverables, then aggregate
  - Clarify what will and will not be done
- Prioritize & Schedule
  - Put first things first (musts vs. wants vs. out-plan)
  - Determine flexibility of functionality vs. schedule vs. resources
  - Develop action items, owners, and due dates
  - Create a schedule with appropriate milestones
  - For large projects or programs, set aggressive and base dates
- Trace & Track
  - Note the source of each requirement and track changes (focus on what/when, as you gather requirements)

Lifecycle

- Remember “ADDIEOs” (pronounce like “adios”)
  - A = analyze the requirements necessary to achieve the objective
  - D = design the product or solution and review that design
  - D = develop the product or solution and inspect/test
  - I = implement the product or solution (put it into service)
  - E = evaluate the product or solution (determine improvements)
  - O = obsolescence planning (migration to new product/solution)
  - s = nothing really – just helps me remember the lifecycle because I can remember how to say “good bye” in Spanish…

Leverage

- Evolve or preserve investments in:
  - People
  - Processes
  - Hardware
  - Software
  - Training

Localization

- Native language support
- Internationalization (I18N)

Usability

- Think ‘customer point of view’
- Users: User-friendly interface
- Developers: e.g., API documentation
- Manageability: ease of maintenance
- Documentation
- Training

Reliability

- Quality
- High-availability (HA) requirements
  - Impact of downtime vs. HA cost

Risk

- Examine risks to project schedule

Performance

- Response time requirements
- Throughput requirements
- Benchmark goals

Price

- ROI
- Consideration of budget cycle
- Make vs. buy (link with strategy)

Politics

- Who are the sponsors?
- Who are the stakeholders?
- What are the positions/hot buttons?
- What are the win/wins?

Support

- Determine customer support requirements
  - How clearly does the functionality report support problems?
  - Support plan
  - Support documentation
  - Do the support requirements match the high-availability requirements?
  - Service Level Agreements (SLAs)

Security

- Authentication
  - Intrusion detection
- Authorization
- Virus detection
- Firewall
- Actions to be taken in case of security violations (e.g., reporting)

Scalability

- Growth
- Capacity (dynamic?)
- Replication

Standards

- Benefits
    - Time to market
    - Investment protection
- List the relevant standards; analyze market trends
- Plans to comply/test

Success Measures

- Prevents scope creep, cost overruns, and customer dissatisfaction
- Define the project-complete deliverables up-front (internal/external)
- Create a checklist with these empirically verifiable items/deliverables and approve it with the customer before the project begins
- At the end of the project and prior to customer review, go over the checklist – make sure the functionality delivers the agreed-upon items
- Review the checklist with the sponsor(s)
- Request project sign-off by sponsor(s)

Templates

- Front-matter
  - Introduction (about you, your company, etc.)
  - Audience
  - Tables (contents, figures, contacts)
  - Legal Terms
- Project Overview
- O’FL^{3}UR^{2}P^{3}S^5
- Project Schedule (e.g., MS/Project)
- Change Control Management Process
- Contingency and Escalation Plan
- Glossary

Case Study #1 (SW Program)

- Interview the customer to determine the requirements
- *A company would like to push product information out to its perspective customers, by their own request, and do this in time for their January product launch.*

- Objective: “To <create a subscription service>, by <January product launch (base date) or pre-holiday season (aggressive date)>, <to proactively keep customers informed of new and existing products>.”
- Functionality: …, Use email for information push, web-enabled customer interface to select products and subscribe/unsubscribe, an administrative interface to provide product information, …
- Lifecycle->Analysis: …, Should we email web pages or URLs in a text message?, …
- Leverage-> … , Company website to provide the customer user interface, …

Case Study #2 (OpenView/HA)

- *From the objective: …[these] mission-critical applications must be “always on:” detect failure and immediately recover …*

- Localization->N/A
- Usability->…, Minimal training required, Access/fix from anywhere, …
- Reliability->…, Mission-critical applications must be ”always on”, …
- Risk->…, One level of redundancy, no UPS for backup routers, …
- Politics->…, [named stakeholders] must buy-in to monitor/recover/risk recommendations, …
- Price->…, Deliver solution within [this] budget, …
- Performance->…, [this] is the time limit from failure detection to recovered mission-critical applications online; recovered mission-critical applications must have [this] response time and [this] throughput, …

Case Study #3 (Merger)

- *From a "merging two companies IT infrastructures" objective: … leveraging company A's worldwide locations and legacy systems and company B's modern IT enhancements …*

- Support-> …, Honor existing IT SLAs across both companies, …
- Security-> …, Add policy-driven intrusion detection and response capability to A's security infrastructure, …
- Scalability->…, A/B company IT infrastructure must support 20 additional sites around the world by 2005, …
- Standards->…, UNIX runs back-office, mission-critical applications, Linux runs web servers, Windows runs front-office, …
- Success Measures-> …, SLAs maintained to [this] level of customer satisfaction as determined with [this] survey, …