

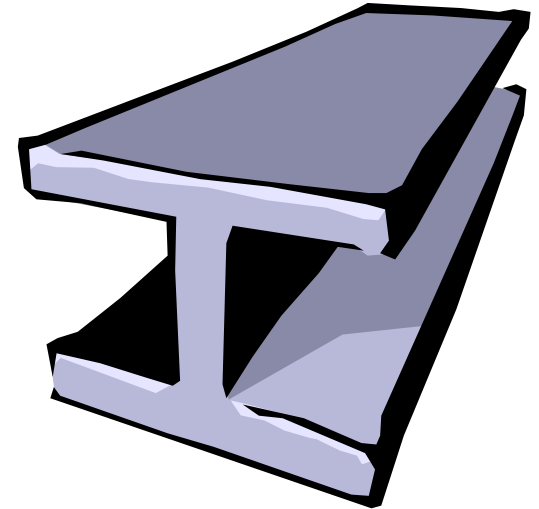
Price to Value

... more than setting a dollar amount

ATC PMM Forum
14 November 2002
Don Jarrell

Imitating a Hard Goods Sale ?

- Enormous familiarity and momentum toward selling goods
- Possession vs Use
- Usually leaves money on the table
- Little relationship building; easy customer attrition; start-over customer acquisition
- Suite & “Bloatware” approaches: different time; different market



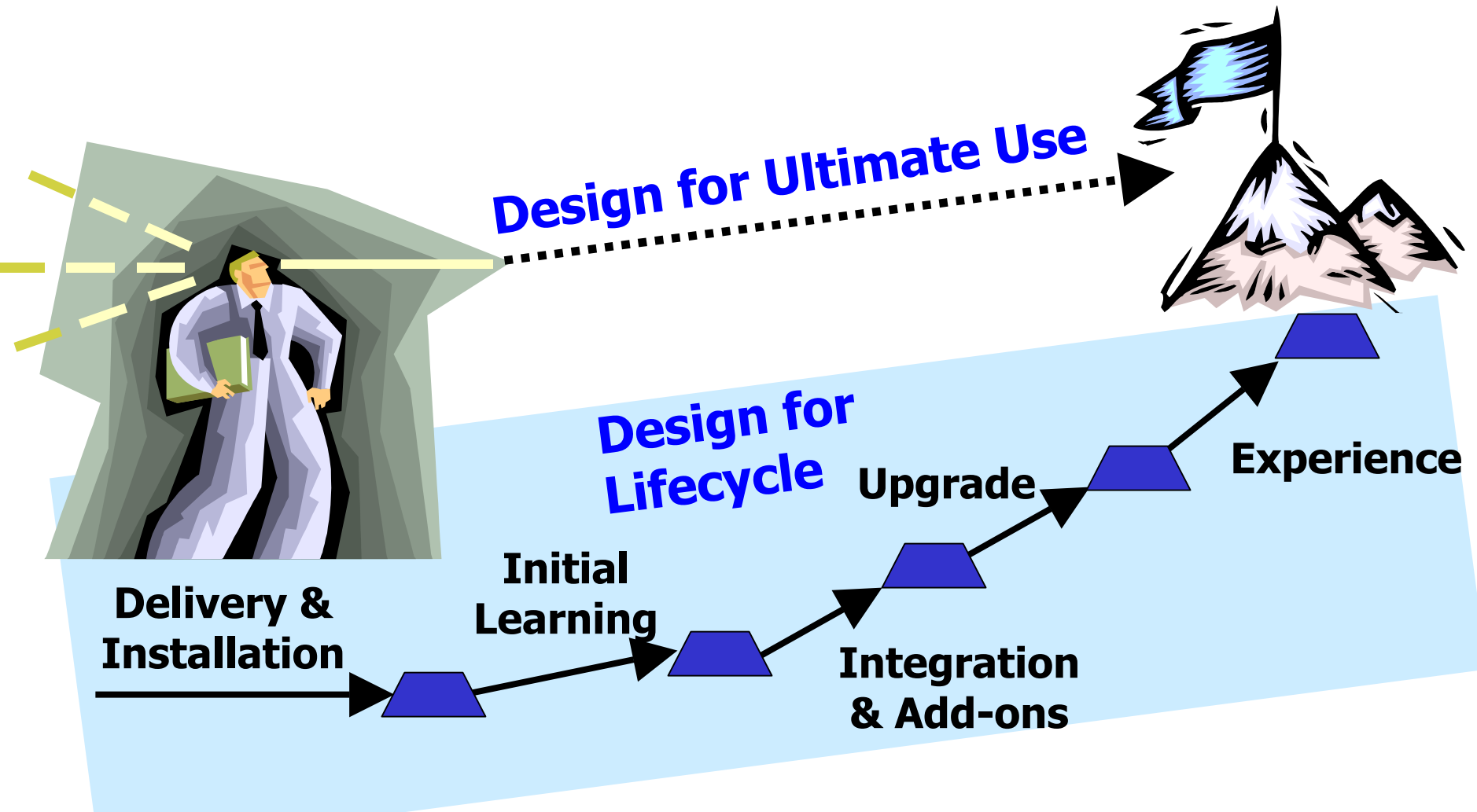


Price-to-Value Defined

Establish reasonable price tags on the product and service elements that represent critical value in the customer's context and collect payment at the time that value is realized.

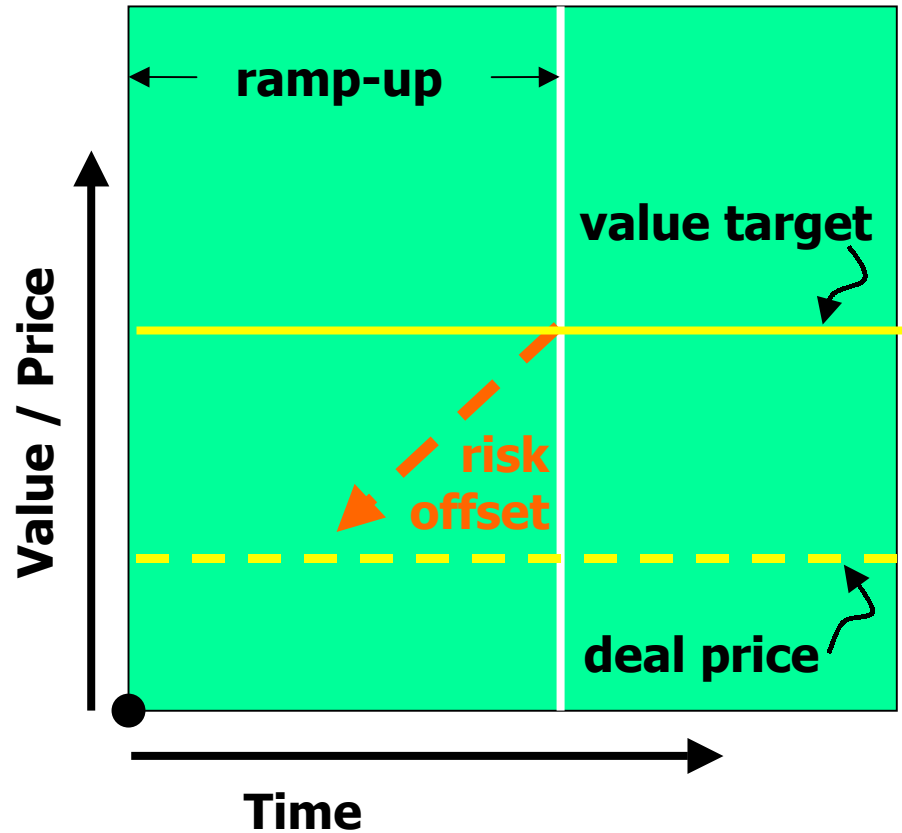


Typical Engineering Bias



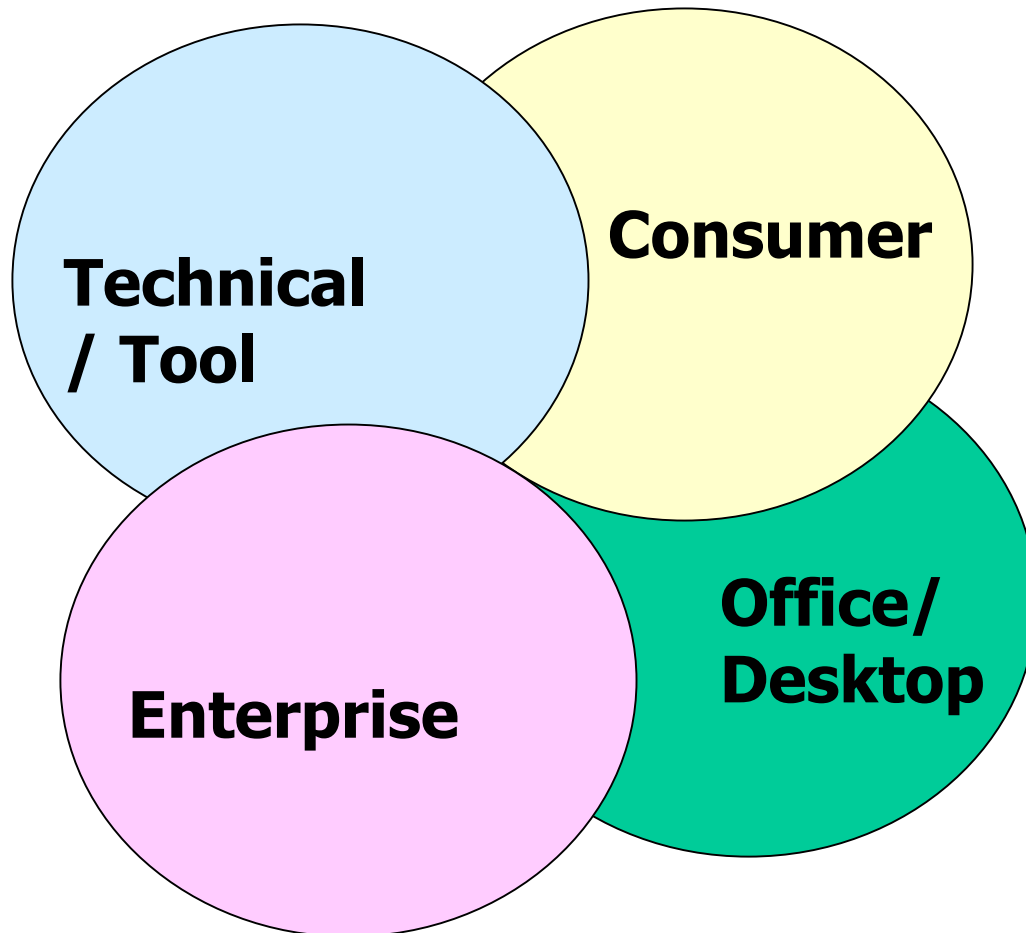
Value Risk Discount

- Value in use achieved after some ramp-up period (learning, growth, rollout, implementation)
- “What-ifs” = risk = $< \$$
- Time-value of money is always applicable



Software Categories

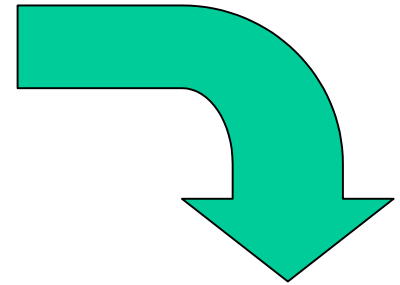
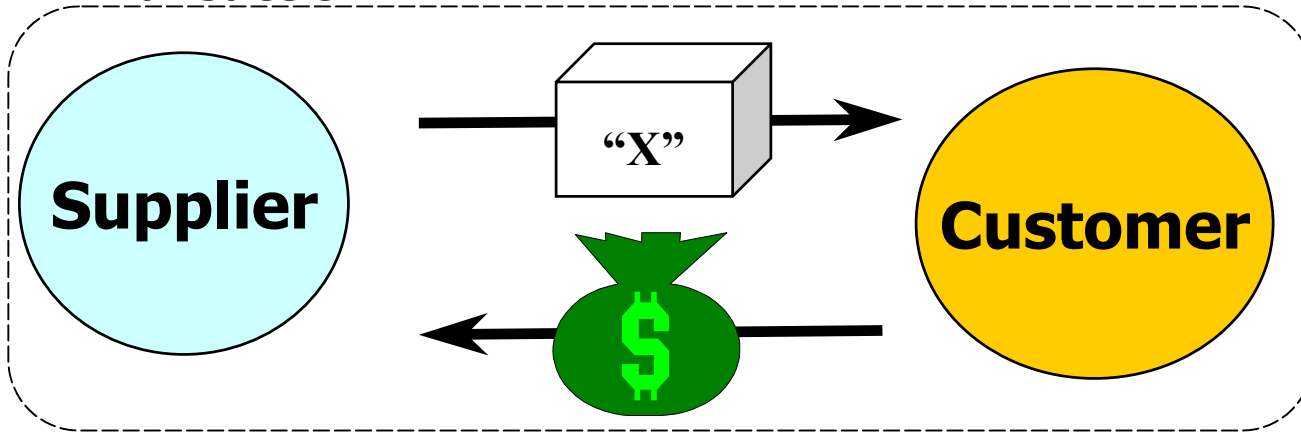
**Value drivers
sometime overlap or
conflict**



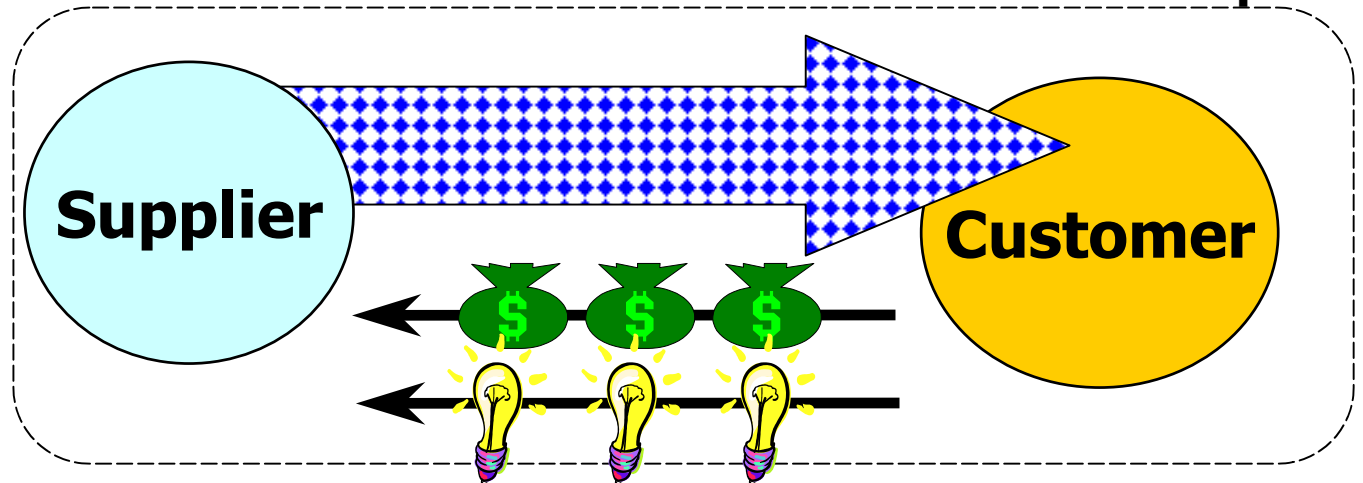
**Important to
know which
key factors
impact your
market and
the decision**

Primary Mindset Shift

Transaction



Relationship



Relationship Factors

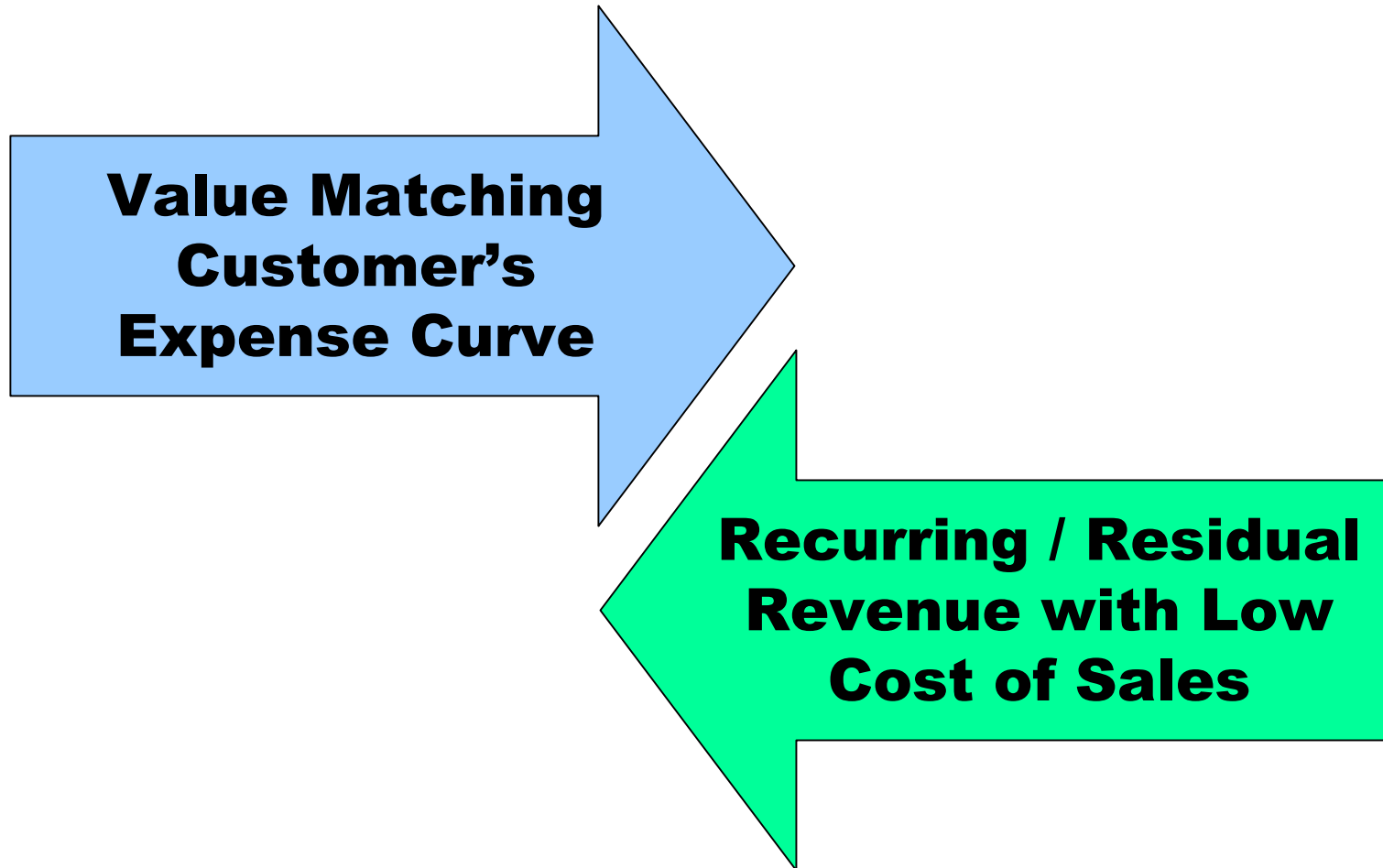
- Total Cost of Ownership
 - Learning (ease, depth, portability)
 - Parallel progress + Leadership
 - Predictability / Roadmap
 - Synergistic value-add
 - E pluribus unum
- ... parts of your brand strategy ???

Who and How ?

- Who is the “customer” ?
- What is the user’s real interaction ?
- Get through to others
 - Min: org, purch mechanics, “what would they say”
 - Better: meet the others
- Aim for evolved organization

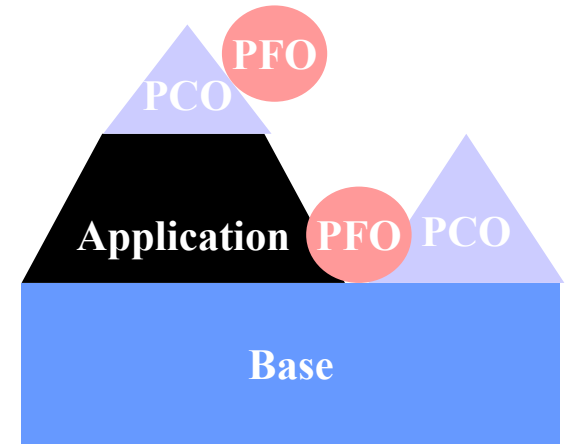


Mutual Benefit



Design for Lifecycle

- Easy/cheap/sublime delivery & payment
- Modularity
 - Building Blocks
 - User-type variants
 - Feature Packages
 - Configuration/capacity options
 - Pre-planned integration & synergies
 - Feature/scale keys



Design for Lifecycle

- Committed tiered service options
- Planned multi-level version stream
 - Necessary fixes (free to everyone)
 - Improvements/ enhancements (small \$ if not on maintenance)
 - Significant new features (more \$)

R1.0

R1.0a

R2.0

R2.1

R3.0

R3.0a

Design for Lifecycle

- Semi-dependent subsequent purchases
 - Subscriptions
 - Add-ons
 - Upgrades
 - Training/Certification (Multiple Tiers)
 - Consumables
 - Lowest acquisition costs
 - Don't overdo (lock-in; price gouging)

Design for Lifecycle

- **POSTULATE INJECTION:**
Customization can get out of control very quickly and kill both your P/L and your brand. Don't do it without proper resources:
- Strong grasp of complete operation
 - Experience
 - Plenty of time to plan very carefully
 - 3 times the budget you think you'll need

Timing

- Early access (advisory/beta) very valuable with mission-critical tools – get your price
- Play to egos **and** avoid over-customization
- Golden Beta (avoid antitrust problems)
- Comprehensive launch (training/doc/support)
- Be honest about launch (late better than bad)
- Version-to-version timing

Bundling

- **Related** to primary product
 - HW - SW
 - Enablers
 - Extensions
 - Same category or theme
 - Training material (standalone quality)
 - ‘Above the line’ (not remedial) services
 - For tools: elemental or sample workproduct (Sample code; clipart; libraries; templates;)

Alliance Promotions

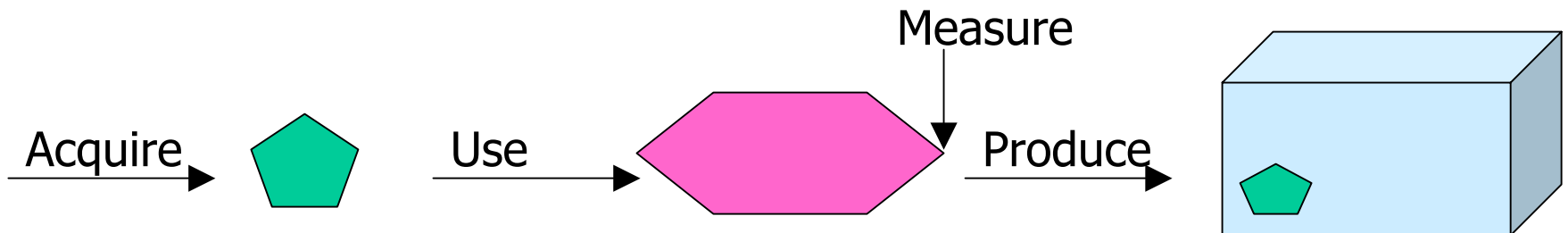
- Similar to bundling
- Borrowed credibility / brand capital
- Bingo cards
- Discount
- Especially helpful with dependent products
 - Precision graphics SW + hi-perf graphics board
 - Compute-intense application + High-end workstation
 - Joystick/console + game software

Free Trial !

- Technically limited trial period
- Shareware costs less but very risky
- Means of distribution critical in setting perceived value of trial product
- Actively survey/convert trial users
- Use with features and modules

Royalty Model

- Pay-As-You-Go - measured use; many forms
- Reduces up-front \$; ties payment to value
- Sublime/easy mechanics are critical
- Psychology issues must be addressed



Open Source Software

- One of three licensed distribution models
- Redefinition of “product” and operation around distinct value proposition
- Licenses still important; service agreement very important (focus of revenue)
- Not for the faint-hearted

Summary

- Design for Lifecycle
 - Granularity, Service (Sublime)
- Timing
- Bundling (products, service, workproduct)
- Alliance promotions
- Try-and-buy / Shareware
- Royalty model
- OpenSource

Interaction

- What about your products ?
 - Software, systems, services
 - Produced by you are acquired by you
- How could you use these techniques ?
- What might you do differently in planning a product or a launch ?

Take-aways

- **Thinking:** Transaction → Relationship
- **Design:** Ultimate Use → Lifecycle
- **Bundling:** Mixed Bag → Related V/A
- **Packaging:** Physical → Logical
- **Delivery:** Large/media → Modular/ESD-Key
- **Payment:** Up-front → Value Realization
- **Service:** Adjunct → Key part of model

... Yours ...

Thank You