

SYSTEMS AND
SOFTWARE
CONSORTIUM

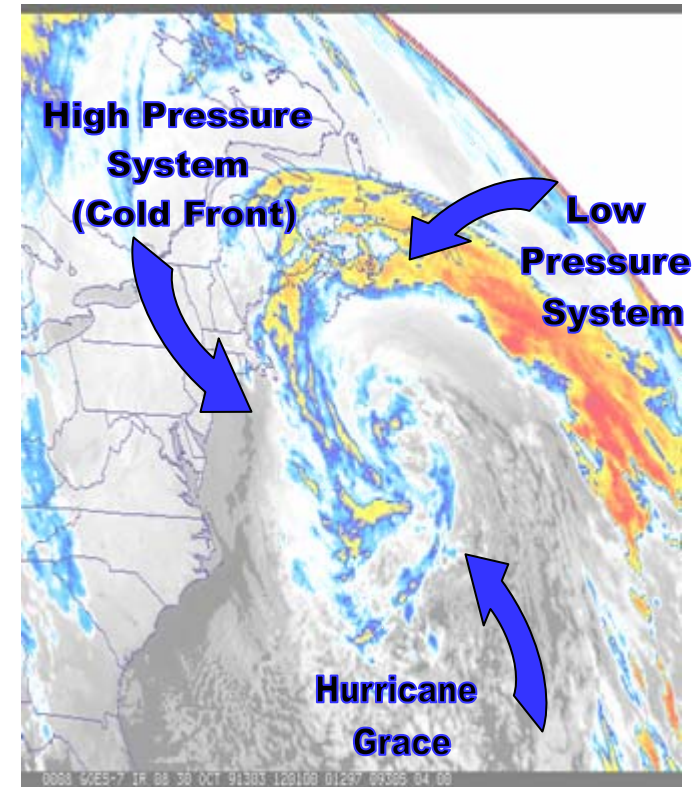
*BUILDING BETTER
SOLUTIONS TOGETHER*

Systematically Tailoring the Organization's Standard Process for a Project

Date: November 16, 2006
CMMI Technology Conference and User Group
Presented By: Dr. Kenneth E. Nidiffer
nidiffer@systemsandsoftware.org
(703) 742-7110

Agenda

- Exponential Growth in Software
 - Expansion in Software Process Assets
- Current State of Practice
 - Large Organizational Standard Processes (OSPs)
 - Manual Tailoring of OSPs to Project Define Processes
- Problems with Manual Tailoring
- Automated Rules-Based Tailoring
- Lessons Learn, Contributions & Road Ahead

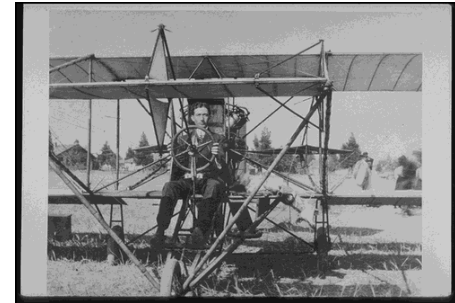
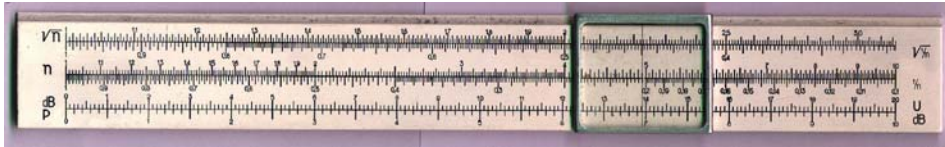


“Perfect Storm” Event, October 1991
National Oceanic & Atmospheric Administration

Exponential Growth in Software Usage & Assets

* Provided by Lockheed Martin

In The Beginning



1960's

1970's

1980's

1990's

2000+



F-4A
1000
LOC

F-15A
50,000
LOC

F-16C
300K
LOC

F-22
1.7M
LOC

F-35
>6M
LOC



Market Dynamics: Drivers That Increase the Demand on Additional Systems & Software Assets

Platform → *Customer Emphasis* → **Enterprise**

Requirements → *Acquisition Model* → **Objectives**

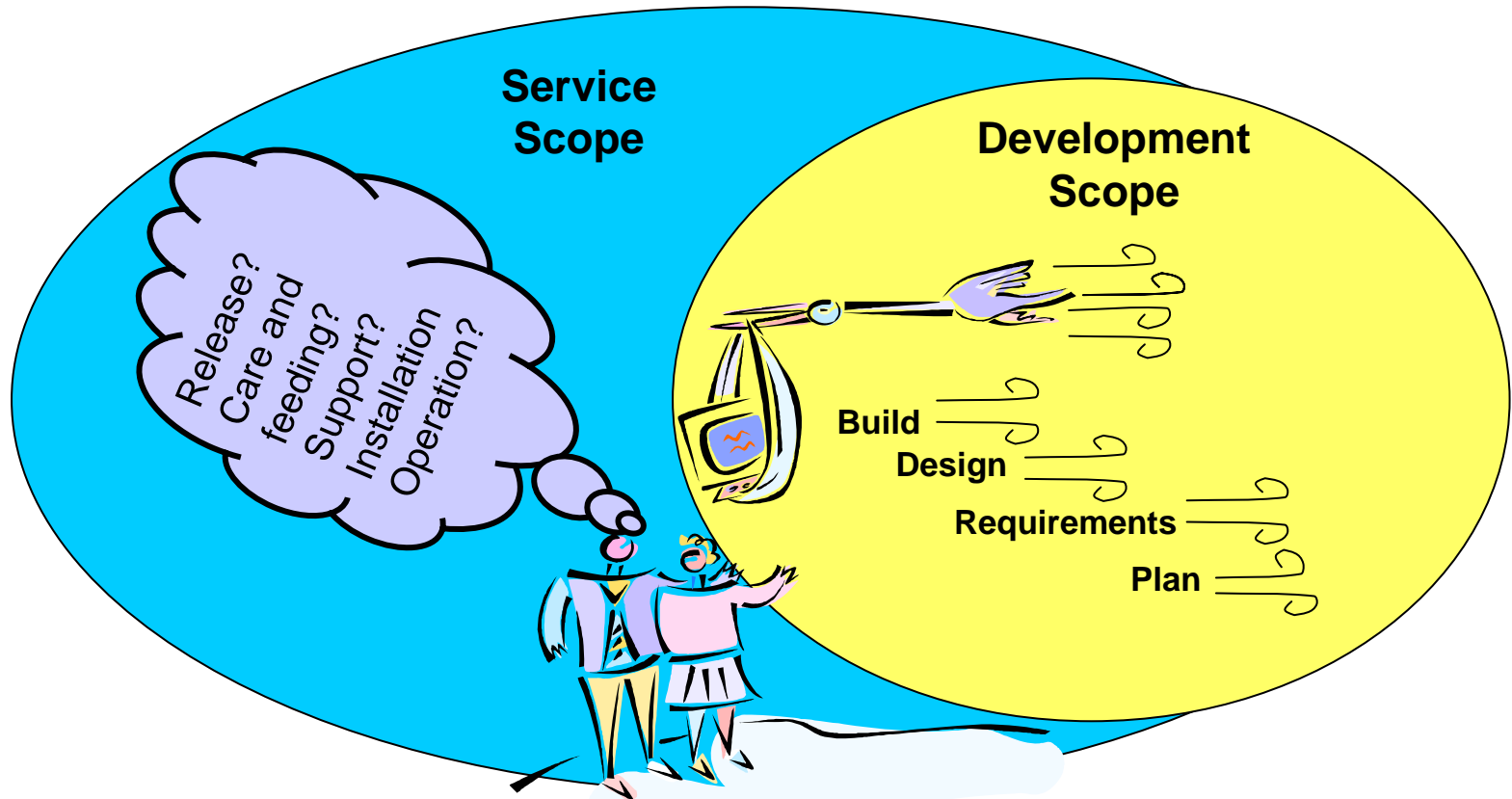
Dominant Prime → *Program Execution* → **Strategic Teaming**

“Boxes” → *Integration Challenge* → **“Layers & Stacks”**

Proprietary → *Architectures and Standards* → **Plug & Play**

The emerging dynamic is to address both sides, and do so with compressed delivery schedules via improvements in systems engineering

Example: Shift to Service Delivery versus Produce and Delivery Perspective

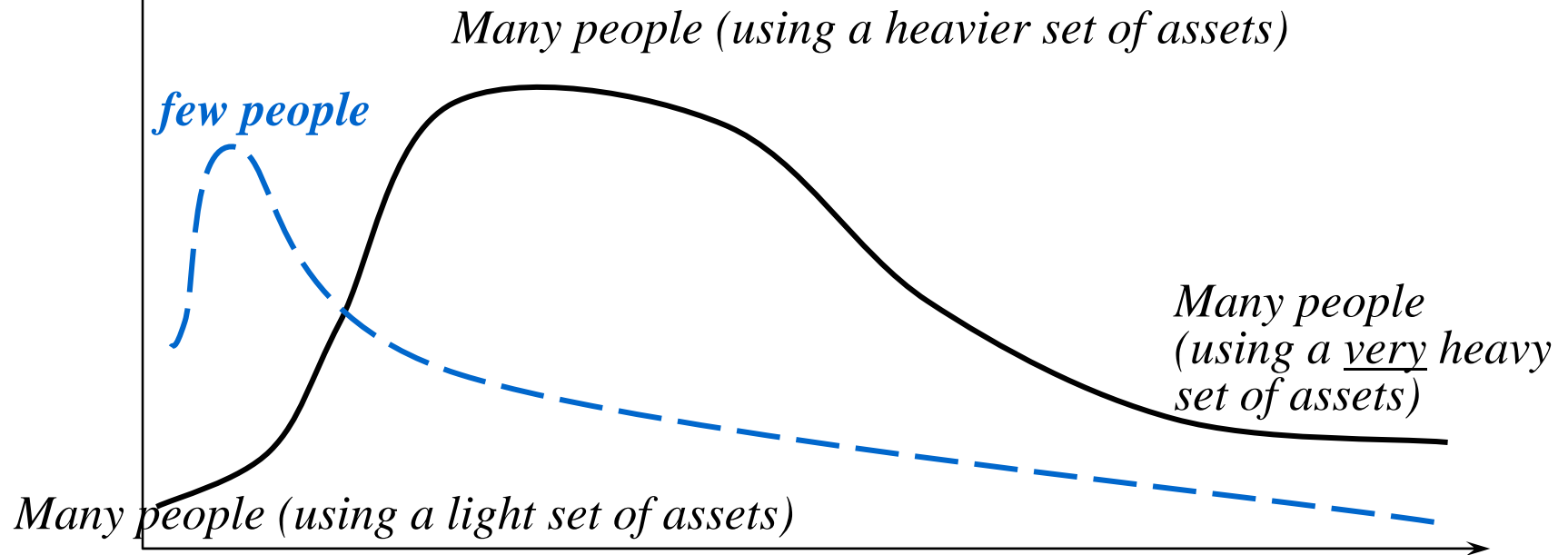


Services represent 80% of the US economy (Source: Paulson, Linda. "Services Science: A New Field for Today's Economy." IEEE Computer Society, August 2006).

Expansion in Process Assets Versus Usage*

What size problem can a given number of people attack, using different sizes of organizational assets?

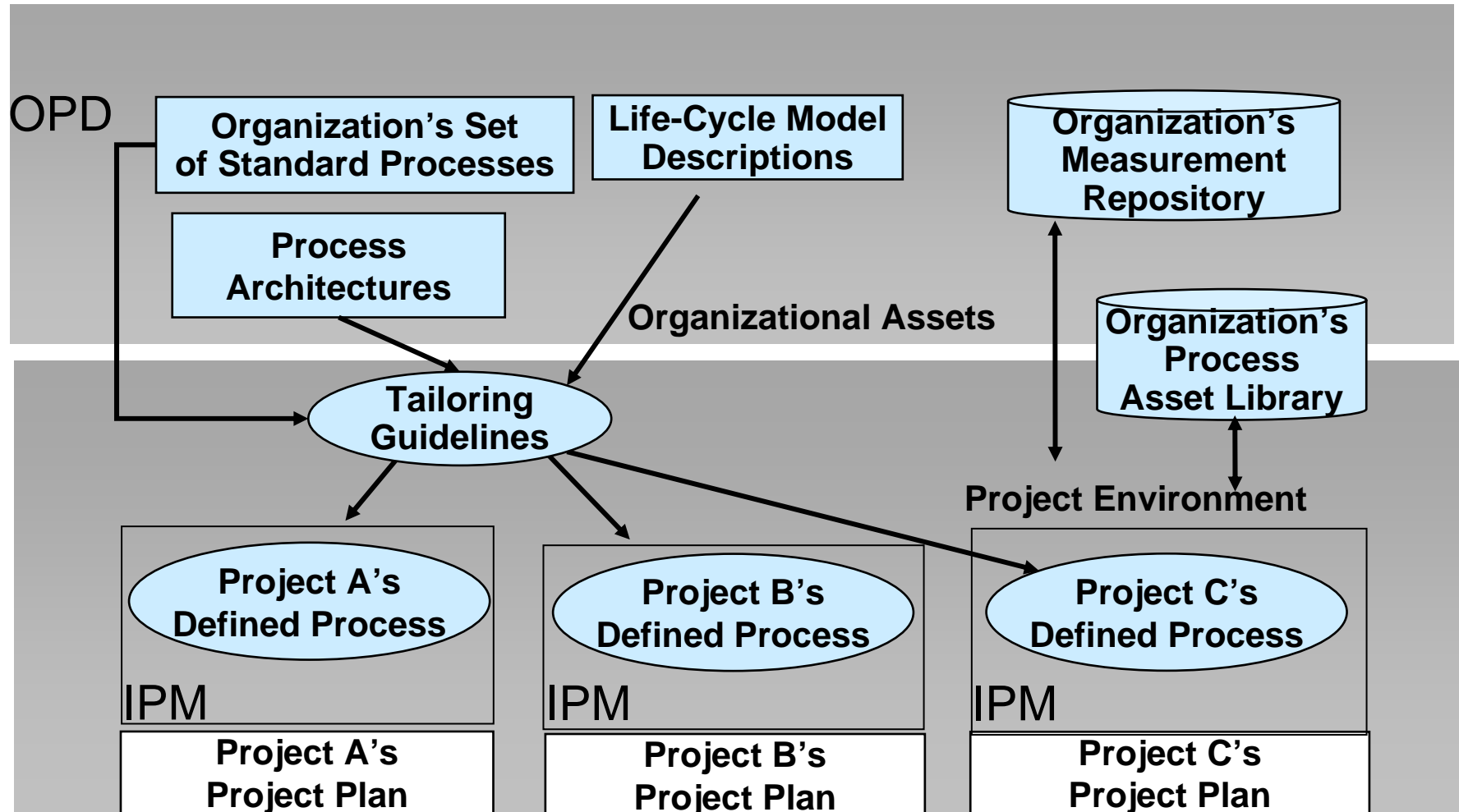
Problem Size



Size of Organizational Assets

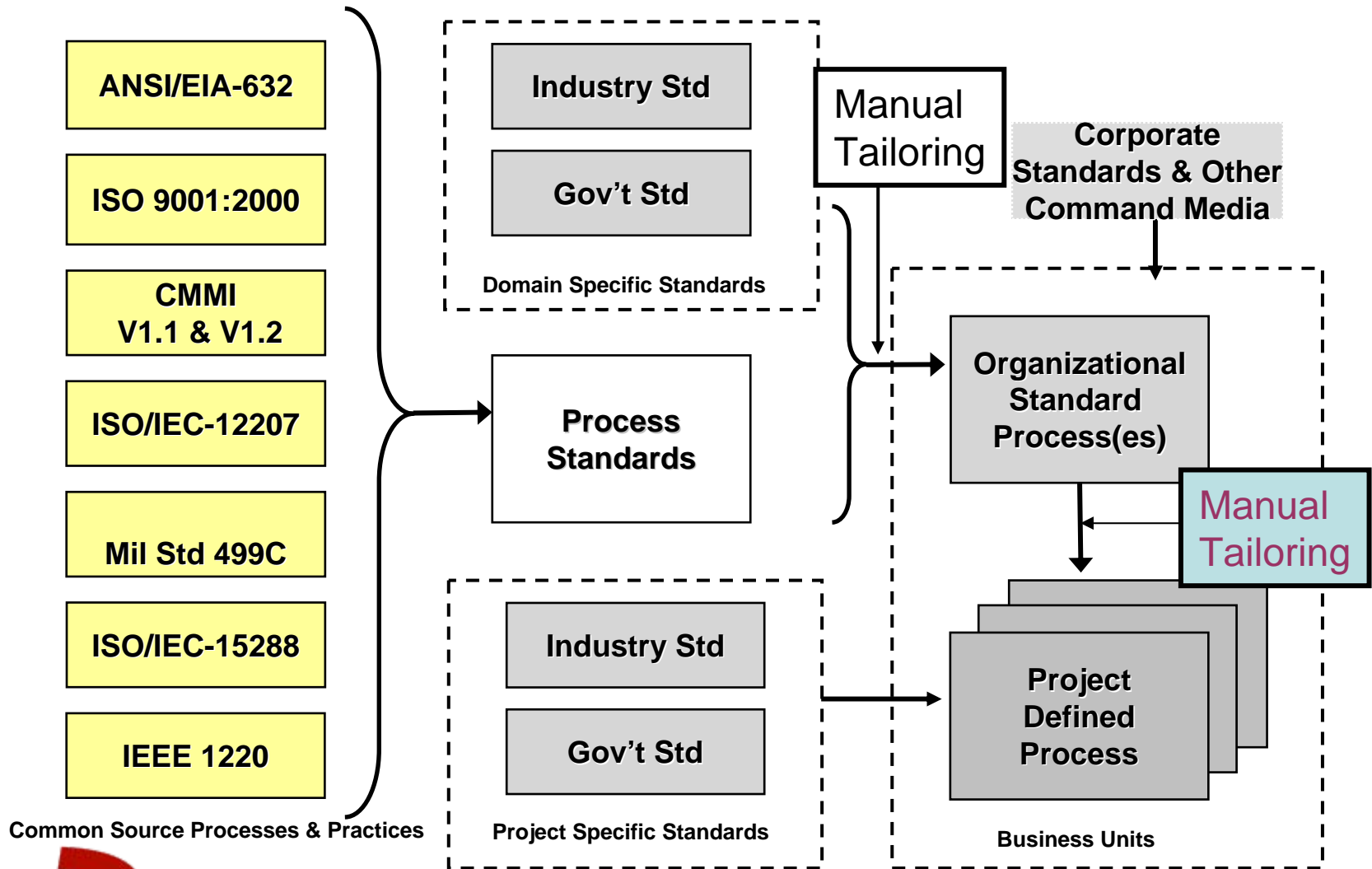
*Slide adapted from Alistair Cockburn, Presentation at SSCI, 3/29/06

Tailoring the Corporate OSPs to Lighter Project Defined Processes

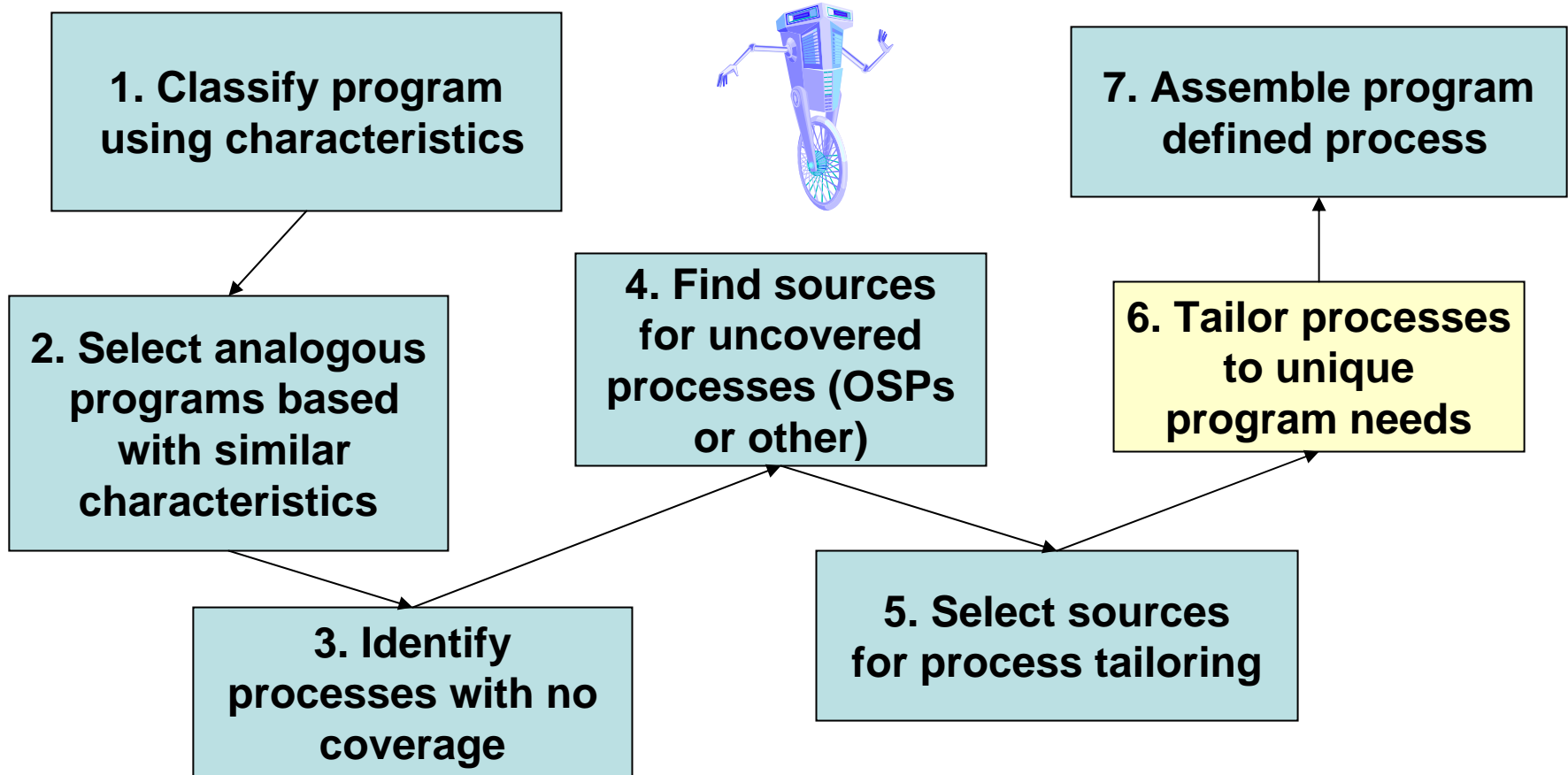


*Training material registered in the U.S. Patent & Trademark Office by Carnegie Mellon University.

Tailoring Example: Software & Systems Engineering Processes & Practices



In Theory, These Steps Should Work:



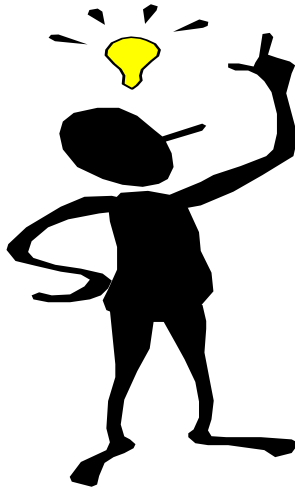
The difference in “Theory” and “Practice” is that in theory they are the same however in practice they often are not

Acknowledged Problems with Manually Tailoring

- Low perceived value by program managers
- Insufficient knowledge and implementation of tailoring rules
- Too much data
 - Organization Standard Process consists of hundreds of process elements; thus, difficult to determine which ones are applicable to their project
- Insufficient planning time available
- Lack of qualified resources
 - Organizations who are lucky enough to have professional members of their technical staff who can manually perform the needed tailoring are often not available

Potential Solution: Automated Rules-Based Tailoring Engine

What is the feasibility to using an intelligent agent (e.g. Tailoring Engine) to inform and direct the selection and tailoring of the OSP based on a set of business rules?

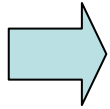


Proof of Concept Approach

- Research the field (please see references)
 - Identified key issues and risks
- Selected a “Proof of Concept” approach
- Develop an engagement plan
 - Defined the problem statement among key stake holders
- Defined the type of OSP we wanted to tailor based on automated process tailoring needs
- Selected a publicly available, accepted and reasonably complete functional process guide as a mock OSP
- Defined the type of tailoring tool we needed based on our tailoring requirements
- Selected the tailoring tool
- Developed a set of tailoring rules for the proof of concept demonstration
- Collected contributions and lessons learned



The Formulation of Effective Tailoring Rules Challenging Due to Project Asset Attributes



Common Source Standards

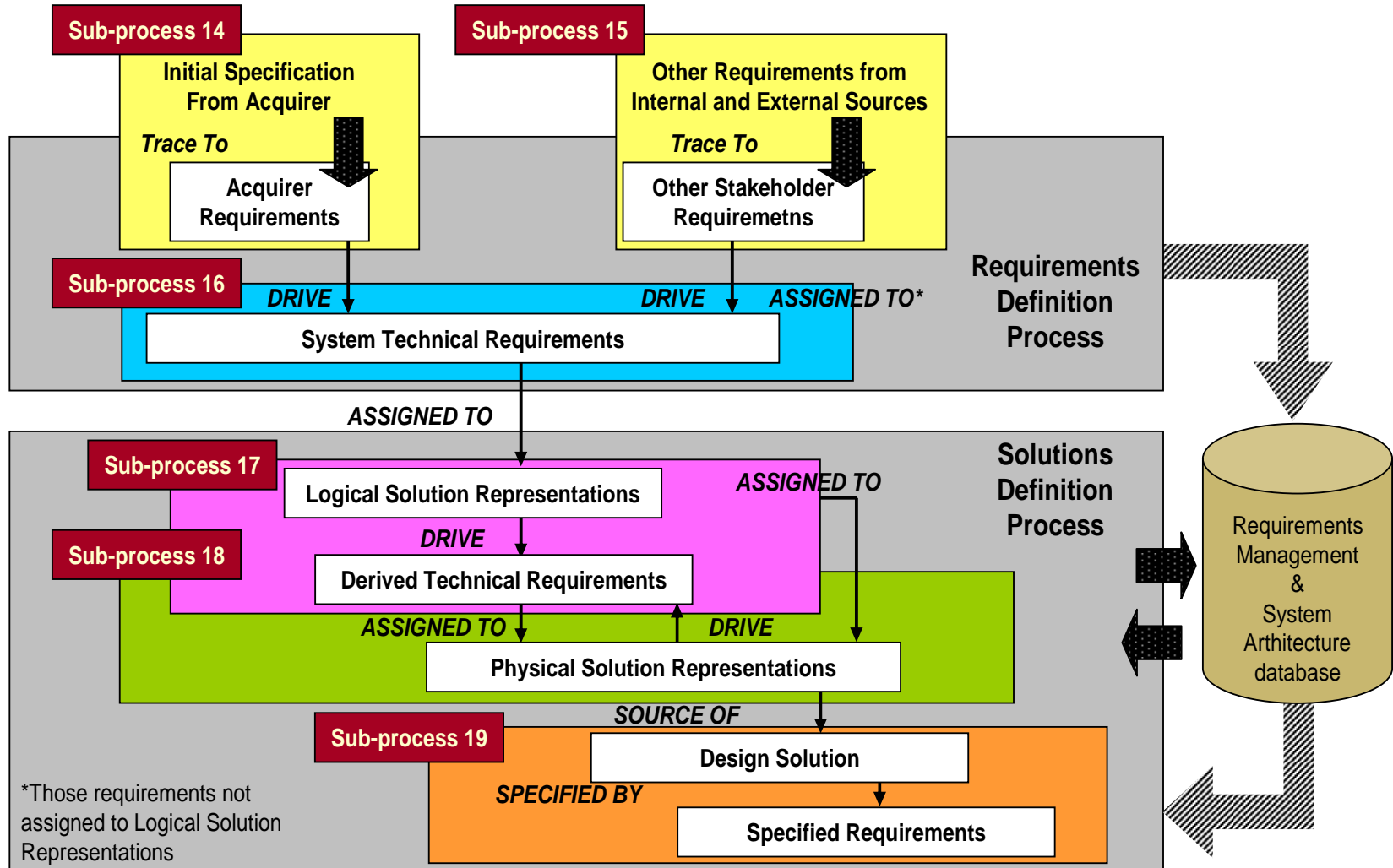
- Project Characteristics
 - Are requirements well known? (Yes: Waterfall, O&M or Incremental life cycles; No: Prototyping, Incremental, or Evolutionary life cycles)
 - Security Requirements?
 - System Size (S, M, L)
 - Level of Risk

- Project Characteristics (cont'd)
 - How complex is the system? (#COTS, #Shalls, #Interfaces, #Users, SOS, #HWCIIs, #SWCIIs, DP rqts)
 - System Composition (HW, SW, both)
 - Product Intent (Feasibility Study, R&D, Operational Program)
 - Contract Type (FFP, T&M, CPFF, Award Fee)
 - Scope (Formality, Control, Structure)

OSP Example: Systems Engineering Standards



Interdependence of Engineering Sub-processes: Naval Systems Engineering Guide and EIA 632



Deceptive Similarity of Industry Standards

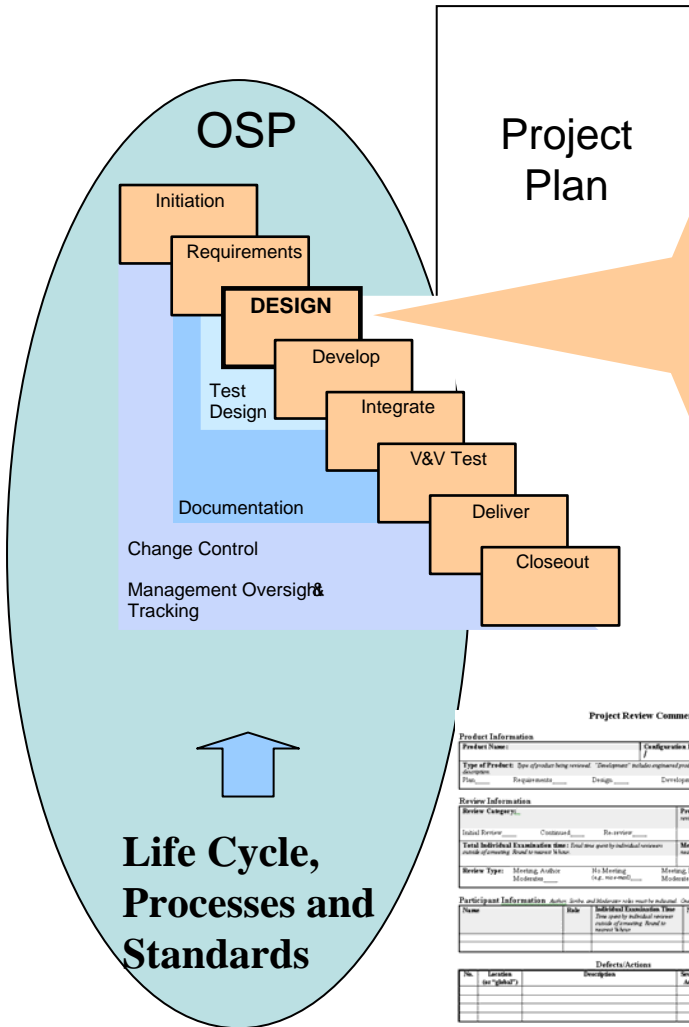
- Because the source standards cover roughly the same ground, we anticipated that “normalizing” them would be relatively straightforward
- We expected to find (and did!) differences in:
 - Scope, terminology and level of detail
- Subtler differences emerged in the source standards’ codification of:
 - Product life cycle
 - Development life cycle model
 - Recursive application of process throughout a system hierarchy
- Industry standards do not cover all the necessary processes (e.g. finance, legal, ethics, etc)

"Inter-connectiveness" of Process Steps

Navy SE Sub-Process	Preceding Process	Next Process	Inputs	Outputs
Acquirer Requirements (SP14)	<p>SP 22 Systems Analysis Process, SP26 Requirements Validation Process</p>	<p>SP 2 Acquisition Process, SP4/5/7 Planning Process, SP 10/11 Assessment Process, SP 12 Control Process, SP 16 Requirements Definition Process, SP22 Systems Analysis Process, SP 26 Requirements Validation Process, SP 31 System Verification Process, SP 33 End Products Validation Process</p>	<p>ICD, CDD/ORD, Engineering Investigation Reports, Utilization & Readiness Reports, Specifications from higher level system building blocks, Sponsor high level operational concept graphic architecture (EXT), Effectiveness analysis reports (SP 22), Effectiveness models (SP 22), Acquirers Requirements Validation Revisions (SP 26)</p>	<p>ICD(SP 2/4/7/10/11/16/3 1/33), Effectiveness Analysis Request (SP 22), Measurement of Effectiveness (SP 5/7/16), CDD or CPD(SP 2/4/7/10/11/16/3 1/33), Specifications from higher level system building blocks (SP 16), Acquirers Requirements (SP 5/16/26)</p>

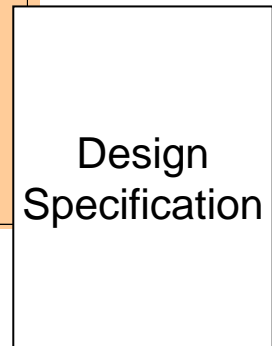
Decomposition - Small Changes in a System Engineering Standard Drive Many Other Changes

Life Cycle Phase Definition



Product Design

Inputs	Entry Criteria	Activities	Exit Criteria	Outputs
Requirements Existing system product documentation	Requirements defined RTM through Requirements Phase updated	Determine Technical Solution <ol style="list-style-type: none"> Using the DAR procedure (TSS, CMMI -PPC-DAR), evaluate and select the technical solution in accordance with the PMR. Using the DAR Procedure, decide whether to make, buy, or re use product components. Update the project plan and WBS, if appropriate. High-level Design <ol style="list-style-type: none"> Develop architecture: Partition product capabilities as environment, product, and interface requirements into components and update the RTM. Collect and analyze existing system product documentation. Develop and maintain the high-level design including interfaces. Peer Review the high-level design using the Peer Review Procedure (TSS, CMMI -PPC-PEER-REVIEW). Review the high-level design with stakeholders, as needed. Detailed Design <ol style="list-style-type: none"> Decompose high-level design into the lowest level product components. Allocate requirements to the lowest level product component, and update the RTM. Define and maintain the control flow of components. Define and maintain the physical structure of components and their relationships. Design inter-component interfaces, user/system interfaces, database, and software/hardware interfaces. Design algorithms and error handling. Perform required data modeling of data and critical computer resource optimization. Complete design documentation. Peer review design components using the Peer Review Procedure. Execute the Baseline Procedure (TSS, CMMI -PPC-BASILINE). Review the final design with relevant stakeholders. Update development folders. Update requirements, operational concepts, and scenarios with information learned during design. 	Design documentation has been baselined	Design documentation Updated RTM Specifications Operational concepts and scenarios



Life Cycle, Processes and Standards

Project Review Comments/Summary Form

Product Information

Product Name: _____ Development Identifier: include date and/or other indication of the version

Type of Product: Use of product being reviewed: Design/development/production such as hardware and software. This includes design of other products upon which this product depends. Requirements _____ Design _____ Development _____ Test _____ Other _____

Review Categories

Initial Review: _____ Continued: _____ Re-review: _____

Product Size: amount of product reviewed _____ Meeting Date: Date of meeting of formal meeting held or date comments due if no meeting

Target Individual Examination Date: Total time spent by individual reviewers _____ Meeting Duration: Time elapsed at the meeting. Leave blank for life sharing review. Baseline meeting 24 hour

Review Types: Meeting, Analysis, Hi Meeting, etc. Hi Meeting: _____ Meeting, independently: _____ Re-review Required?: Yes _____ No _____

Participant Information Author, Scribe, and Chairperson roles must be indicated. One individual may have multiple roles.

Name	Role	Individual Contribution Title	Date

Defects/Actions

No.	Location (in "ghidra")	Description	Severity/Action	Type	Issue Number	Issue Number	Issue Number	Comments

Peer Review Procedure

Purpose
This procedure provides a consistent means of conducting peer reviews for work products.

Responsibility/Authority
Author

Special Circumstances
This type of peer review is defined which vary in formality. The type of peer review you select should be based on the risk and complexity of the work product being reviewed in addition to the resources available. Types of peer review are: checklist, individual, and team. Individual review is conducted on documents, and generate signed acceptance data. The type of procedure used for this review is defined in table following the procedure type.

Inputs

- Work product
- Applicable base supporting work product

Exit Criteria

- Completion of work product
- Production work product is available and has been reviewed (if applicable)

Output

- Completed Peer Review
- Individual Contribution Form
- Individual Contribution (if applicable)

Activities

Preparation for Peer Review

After determining that the entry criteria have been met, prepare review packages.

NOTE: The reviewed work product is a document, spreadsheet, or identify completion items needed to provide sufficient information to reviewers. Include the Review Comments/Summary Form. If reviewers will be required to review their comments on the form, the process requires working paper in the process, be sure that the work product version is the version reviewed in the relevant baseline.

Step 1: Identify qualified review participants and assign roles.

Step 2: Determine necessary review guidelines. Identify review participants of date, time, and location of the review. Type of peer review, participant roles, review criteria, review steps, and any procedural information.

Template

Forms

Procedures

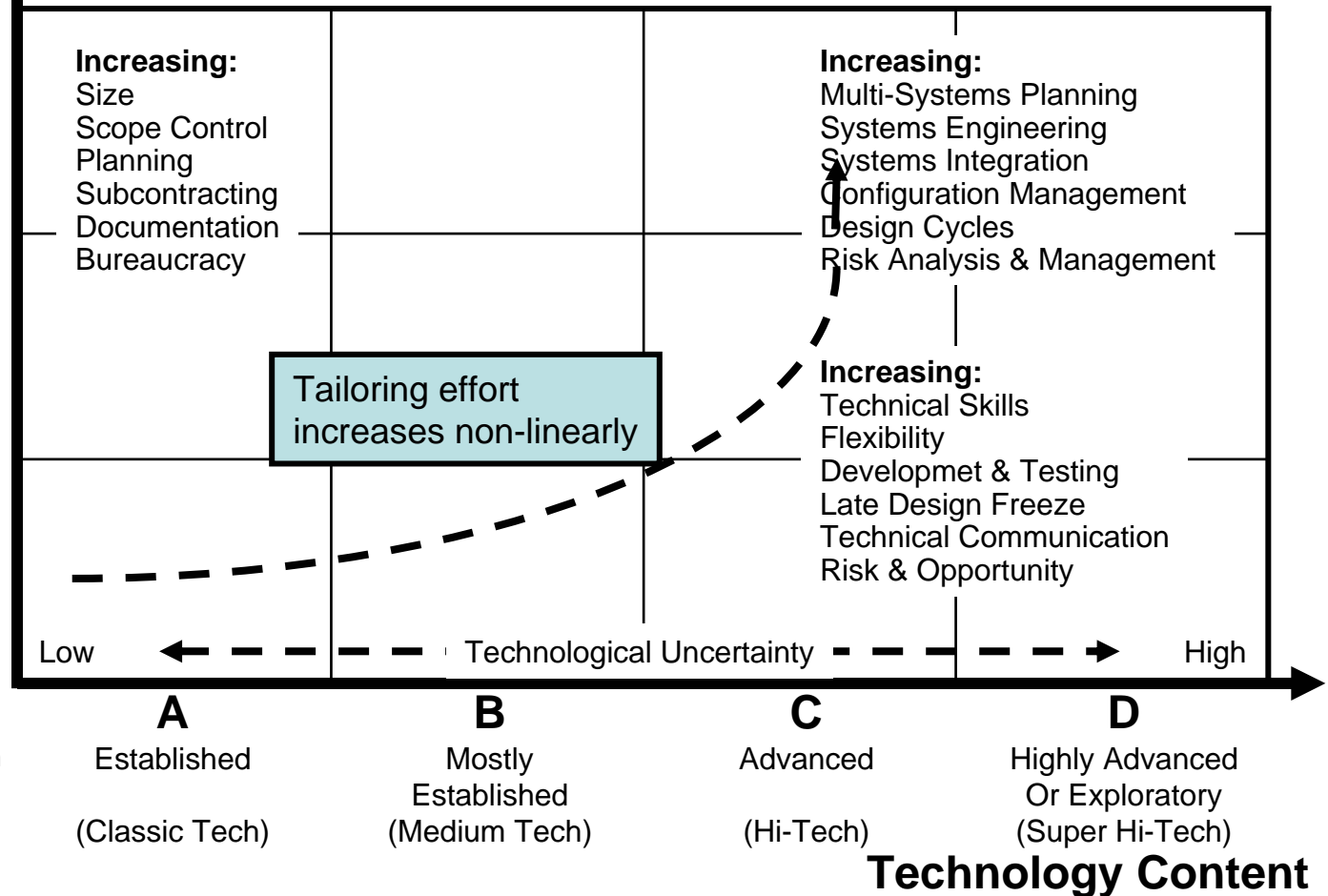
Tailoring Effort Increases Non-Linearly with Program and Technology Complexity

Program/Project Management Scope

3. Array
Program or Set of projects

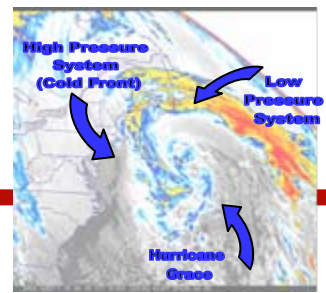
2. System
Project with complex set of interactive elements

1. Assembly
Project consisting of single unit



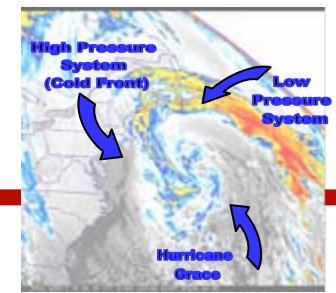
Source: Shenhar and Wideman

What We Determined: Devil in the Details



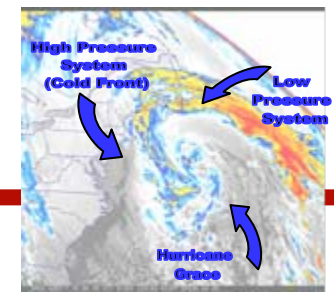
- State of Practice
 - Some on-going research in the area of automated rules-based test engines
 - Existence of a limited set of reasonably mature automated tailoring tools
 - Significant manual-tuning required dealing with most problems
 - Limited number of “universal” tailoring rules
 - Output often compatible with Microsoft tools
 - WBS and Project Plans automatically generated
 - OSPs nominally contain a substantial body of knowledge and data
 - Many different formats and types of artifacts

What We Determined: Devil in the Details



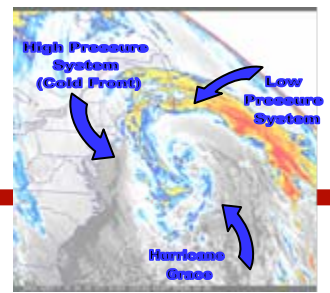
- Technically
 - In general, standard processes are not engineered/architected for tailoring
 - Makes automation difficult
 - Organizational assets are expanding, costly to maintain, and are becoming more difficult to tailor to projects
 - Difficult for an OSP to cover every case and be lean
 - Tailoring rules difficult to establish
 - Numerous attributes drive the process tailoring decisions
 - Strong demand for the establishment of robust tailoring rules

What We Determined: Devil in the Details



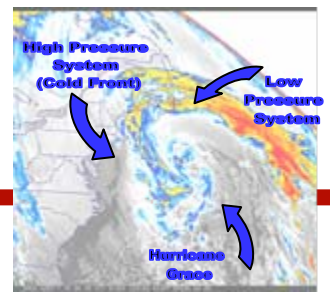
- Technically (Con't)
 - Multiple frameworks (e.g. CMMI, ISO, ITIL, etc) are not scalable or easily customizable
 - Different frameworks need different data elements
- Organizationally
 - Corporate committed, projects involved
- Human element
 - Natural tendency to tailor the OSP based on least resistance
 - Information from similar programs
 - Reliance on local process improvement teams
 - Difficult to demonstrate to program managers the ROI
 - Lack of consistency in process definition for most process areas

Road Ahead



- The problem is real
 - The problem is not going away and will grow in importance with time
- In the short run, tailoring responsibility will nominally rest with the process engineering teams
 - Key focus will be the development of tailoring rules and guidelines
 - Rules-based engine research will continue
- In the longer term:
 - Organizations will use process architectures to design and populate their process asset libraries
 - Intelligent agents/rules-based test engines will become more prevalent

Road Ahead



- Utopia Perspective
 - What could we do differently if we started over?
- Secret of success may be a shift of focus from assets to process execution information
- Approach
 - Define a common set of software and system engineering processes for the organization
 - Map the common set of processes to the organizations lifecycles
 - Apply processes/lifecycles on programs and collect data
 - Store process execution information in the PAL
 - Improve processes based on results and establish tailoring rules

Road Ahead: Example - Service-Oriented Architecture & Asset Management

- Service-Oriented Architecture is an approach to building IT systems out of common parts
 - Represents a breakthrough in the way we build IT systems
 - Composed of reusable components, called services
 - Service is a building block that performs a distinct function
 - Evolution of client/server architecture
 - Functions of user I/F, application logic and data management are separated and decomposed still further

• Why Now

- Internet and World Wide Web →
- Business/Quality Focused →
- Standardization (common parts) →
- Vender Market →





Questions?

References

- Brown, J.S., Duguid, P., "Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation." *Organization Science*, 2 (1), 1991, pp. 40-57
- Busak, Mike, "Achieving CMMI ® Compliance through Automated Process Management", White Paper, SELECT Business Solutions, 2004
- Cockburn, A., "Selecting A Project's Methodology." *IEEE Software*, 14 (4), 2000, pp. 64-71.
- Dybå, T., "Improvisation in Small Software Organizations." *IEEE Software*, 17 (5), 2000, pp. 82-87
- Cockburn, A, Presentation at SSCI, March 2006

References

- Fischer, G., Ostwald, J., "Knowledge Management: Problems, Promises, Realities, and Challenges." *IEEE Intelligent Systems*, 16 (1), 2001, pp. 60-72
- Henninger, Scott, "Turning Development Standards Into Repositories of Experiences, Department of Computer Science and Engineering, University of Nebraska, NE
- Henninger, Scott, "Tools Supporting the Creation and Evolution of Software Development Knowledge, Department of Computer Science and Engineering, University of Nebraska, NE
- ANSI/EIA Standard 632, Processes for Engineering a System, January 1999

References

- ISO/IEC Standard 15288, Systems Engineering – System Life Cycle Processes, 2002
- EIA Standard 731.1, Systems Engineering Capability Model, August 2002
- IEEE 1220 Standard, IEEE Standard for Application and Management of the Systems Engineering Process
- Naval Systems Engineering Systems Guide, October 2004
- INCOSE Systems Engineering Handbook, Version 3, June 2006

Recommended Reading

- Buckman, Robert H. *Building a Knowledge-Driven Organization*. McGraw-Hill, New York, NY, 2004.**
- Chao, Pierre A. "Alternative Futures for the Defense Industry." Center for Strategic & International Studies, Chantilly, VA, April 2005.**
- Chesbrough, Henry William. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Publishing Corporation, Boston, MA 2003.**
- Drucker, Peter. *Managing in the Next Society*. Truman Talley Books, New York, 2003.**
- Gladwell, Malcolm. *The Tipping Point*. Little, Brown and Company. New York, 2002**
- Malone, Thomas. *The Future of Work: How the New Order of Business Will Shape Your Organization, Your Management Style and Your Life*. Harvard Business School Publishing, Boston, MA, 2004.
See <http://ccs.mit.edu/futureofwork/>**
- Wladawsky-Berger, Irving. "The Future of IT in an On-Demand World." IBM Server Group, Keynote address at OSBC 2005.
Archived at <http://www.itconversations.com/shows/detail495.html>**